Time-Slicing, Rescaling & Ratio-based Parallel Time Integration

Noha Makhoul-Karam

N^o d'ordre : 4312 ANNÉE 2010

THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Européenne de Bretagne

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MATISSE

présentée par

Noha MAKHOUL-KARAM

préparée à l'unité de recherche 6074-IRISA
Institut de recherche en informatique et systèmes alèatoires
IFSIC

Time-Slicing, Rescaling & Ratio-based Parallel Time Integration

Calcul en Tranches de Temps, Redimensionnement & Schéma Parallèle en Temps par la Méthode des Ratios

Thèse soutenue à l'IRISA
le 17 Décembre 2010

devant le jury composé de :

Damien TROMEUR-DERVOUT
Professeur à l'Ecole d'Ingénieurs Polytechnique de Lyon 1 / rapporteur

Thierry COUPEZ
Professeur à l'Ecole des Mines- Paris / rapporteur

Bernard PHILIPPE
Directeur de recherche émérite INRIA / examinateur

Jocelyne ERHEL
Directrice de recherche INRIA / directrice de thèse

Nabil NASSIF
Professeur à l'Université Américaine de Beyrouth / co-directeur de thèse

A Nagib, mon père

&

A Abdo, mon mari...

Ce travail a reçu un soutien financier de la part de:

1. **INRIA - FRANCE** (Programme SARIMA & Projet SAGE)

2. **CNRS - LIBAN**

Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidée à concevoir, à réaliser et à finaliser ce travail que ma formation initiale en génie civil, et mes dix années d'activités professionnelles en études de structure, ne laissaient point prévoir, mais que mes quinze années suivantes d'enseignement et de pédagodie mathématique ont certainement encouragé.

Je m'adresse en tout premier lieu au Professeur Nabil Nassif. Après avoir dirigé mon mémoire de DEA, il a motivé cette thèse et en a initié la conception et les recherches. Il m'a fait découvrir le plaisir de la recherche et m'a appris à y mettre la rigueur et la technique nécessaires. Il a accepté toutes mes contraintes et m'a procuré un soutien infaillible au cours de ces cinq années. Sa disponobilité constante, ses connaissances mathématiques et son professionalisme certain m'ont permis de mener à terme cet ouvrage.

Je remercie également, et tout autant, ma directrice de thèse, Jocelyne Erhel. En dépit des nombreuses contraintes de temps et d'espace, elle a réussi à encadrer mon travail, me faisant ainsi profiter de sa grande expérience dans le monde de la recherche, de sa rigueur absolue et de tout son savoir-faire. Durant cinq étés consécutifs, elle m'a accueilli au sein de son équipe, mettant à ma disposition toutes sortes de facilités. Il lui est même arrivé de m'offrir l'hospitalité de sa maison (merci Yves aussi) !

Je citerai aussi une troisième personne sans laquelle cette thèse n'aurait pas vu le jour: le directeur de recherche émérite, Bernard Philippe. C'est lui qui m'a encouragée dès le début, m'a mise en contact avec son équipe au sein de l'IRISA et a rendu possibles mes séjours en France grâce au support financier de SARIMA. Lors de mes passages à Rennes, il n'hésitait pas à m'offrir ses conseils, me faisant ainsi profiter de sa grande expérience de chercheur.

Je tiens par ailleurs à remercier Messieurs les Professeurs Damien Tromeur-Dervout et Thierry Coupez d'avoir accepté de juger ce travail et pour l'intérêt qu'ils y ont porté.

J'adresse de grands remerciements à Yeran Soukiassian et Jessy Haikal qui ont collaboré à cette recherche dans le cadre de leur mémoires de mastères. Leur travail efficace a contribué indéniablement à l'avancement des travaux. Merci Jessy d'avoir été là jusqu'au bout et de m'avoir tant aidée sur le plan technique, qui est ton fort. Merci pour ton amitié.

Je remercie toute l'équipe SAGE qui m'a acceuillie tous les étés, depuis juillet 2005. Je me sens faire partie de votre équipe et c'est avec un pincement au coeur que j'entrevois la fin de l'aventure...

Je remercie à présent tous mes proches, en particulier tous les membres de ma famille: ma mère, mes frères et mes soeurs qui m'ont procuré leur support et leur soutien, chacun à sa façon, mais tous avec le même amour...

Finalement, je m'adresse à mes trois filles: Layal, Manal et Lama.
Vous avez été à mes côtés pendant tout mon travail et vous avez contribué par vos sacrifices, votre amour et votre présence, à sa réalisation.
Vous m'avez vue moins disponible... mais je m'en console puisque je n'ai fait que vous donner l'exemple d'un travail accompli avec passion.
Je vous aime tant...

Abstract

Recently, many parallel-in-time algorithms have been proposed for solving initial value problems of the form (S): $\frac{dY}{dt} = F(Y), \quad Y(t_0) = Y_0$, that could follow, for example, from the space semi-discretization of partial differential equations. Since there is no natural parallelism across time, those algorithms are mainly meant to tackle real-time problems or to be superposed to parallelism in the space or the method directions, thus enabling a more effective use of a higher number of processors.

In this thesis, we propose a Ratio-based Parallel Time Integration (RaPTI) algorithm for solving (S) in a time-parallel way, in the case where the behavior of the solution is known. A sliced-time computing methodology underlies this new approach. It consists of (i) a time-slicing technique that ends a slice by shooting a relevant end-of-slice condition related to the behavior of the solution and (ii) a rescaling technique that changes both the time-variable and the solution, setting them to 0 at the beginning of each time-slice. Thus, solving (S) becomes equivalent to solving a sequence of initial value "shooting" problems, in which one seeks, on each time-slice, both the solution and the end-of-slice time.

RaPTI algorithm uses this methodology, and some resulting similarity properties, for generating a coarse grid and providing ratio-based predictions of the starting values at the onset of every time-slice. The correction procedure is performed on a fine grid and in parallel, yielding some gaps on the coarse grid. Then, the predictions are updated and the process is iterated, until all the gaps are within a given tolerance.

The originality of RaPTI algorithm lies in the fact that the predictions it provides, and their update at each iteration, do not require any integration on the coarse grid, unlike other parallel-in-time schemes. Moreover, it does not start with the choice of a coarse grid, it rather starts by choosing an end-of-slice condition that will generate a coarse grid that suits the behavior of the solution.

RaPTI algorithm is applied, in this thesis, to three problems: a membrane problem, a reaction-diffusion problem and a satellite trajectory in a J_2-perturbed motion. In some rare cases of invariance, it yields a perfect parallelism. In the more general cases of asymptotic and weak similarity, it yields good speed-ups.

Keywords

Ordinary differential equations; initial value problems; parallel time-integration; end-of-slice condition; rescaling methodology.

"Les phénomènes qui nous paraissent successifs dans le temps
sont en réalité donnés tous à la fois,
en dehors du temps"

[...] Contre la linéarité du temps,
selon Alfred Jarry (1873-1907), élève d'Henri Bergson

Résumé

Récemment, beaucoup de schémas parallèles en temps ont été proposés pour résoudre des problèmes à valeur initiale de la forme (S): $\frac{dY}{dt} = F(Y),\ Y(t_0) = Y_0$, pouvant résulter, par exemple, de la semi-discrétisation en espace d'équations aux dérivées partielles. L'intégration des problèmes d'évolution étant naturellement séquentielle, de tels schémas ont pour principale motivation de résoudre des problèmes en temps réel ou de se superposer à un parallélisme à travers l'espace ou la méthode et ce, pour un usage plus efficace des machines fortement parallèles disponibles actuellement.
Dans cette thèse, nous proposons l'algorithme RaPTI, qui permet de paralléliser en temps la résolution du problème (S), lorsque le comportement de la solution est connu. Une méthode de calcul par tranches est à la base de ce nouveau schéma. Elle consiste en (i) une procédure de génération de tranches de temps basée sur une condition de fin de tranche que la solution doit vérifier et (ii) un redimensionnement de la variable temps et de la solution qui les initialise à zéro au début de chaque tranche de temps. Ainsi, la résolution de (S) devient équivalente à la résolution d'une suite de problèmes redimensionnés à valeur initiale et avec condition de fin de tranche, donnant à la fois la solution sur chacune des tranches et le temps de fin de tranche.
L'algorithme RaPTI utilise donc cette méthodologie, et certaines propriétés de similarité qui en découlent, pour générer la grille de temps grossière et fournir des prédictions au moyen d'une méthode de ratios. La procédure de correction se fait ensuite, sur une grille de temps fine, en résolvant en parallèle les systèmes redimensionnés. Ceci conduit à des sauts sur la grille de temps grossière. Les prédictions sont alors corrigées et le processus est itéré jusqu'à ce que tous les sauts soient inférieurs à une certaine tolérance.
L'originalité de l'algorithme RaPTI réside dans le fait que les prédictions, et leur corrections à chaque itération, ne nécessitent aucune résolution sur la grille de temps grossière, contrairement aux autres schémas parallèles en temps. De plus, il ne commence pas par un choix de la grille grossière, mais plutôt par le choix d'une condition de fin de tranche qui va générer cette grille de façon bien adaptée au comportement de la solution.
L'algorithme RaPTI est appliqué, dans cette thèse à trois problèmes: un problème de membrane, un problème de réaction-diffusion et un calcul de trajectoire de satellite dans un mouvement perturbé en J_2. Dans quelques rares cas d'invariance, il conduit à un parallélisme parfait. Dans les cas plus courants de similarité asymptotique ou faible, il donne de bons speed-ups.

Mots-Clés

Equations différentielles ordinaires; problèmes à valeur initiale; calculs parallèles en temps; condition de fin de tranche; méthode de redimensionnement.

"Une minute affranchie de l'ordre du temps a recréé en nous, pour la sentir, l'homme affranchi de l'ordre du temps."

A la recherche du temps perdu, Le Temps retrouvé (1927)

Marcel Proust

Résumé Détaillé

On considère dans cette thèse le problème de Cauchy autonome:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t > 0, \\ Y(t_0) = Y_0, \end{cases}$$

dont l'existence et l'unicité de la solution $Y : [0, \infty) \to \mathbb{R}^K$ est supposée bien établie.

Depuis l'avènement d'ordinateurs massivement parallèles, de nombreuses méthodes parallèles sont proposées pour une résolution efficace de ce problème. Certaines de ces méthodes parallélisent les calculs à travers le temps, en fournissant aux divers processeurs des prédictions des valeurs de la solution à leurs temps initiaux respectifs.
C'est dans ce cadre que se place cette thèse dans laquelle nous proposons un schéma parallèle en temps pour la résolution de (S), basé sur une méthode de calcul en tranches de temps, avec redimensionnement.

I- Calcul en Tranches de Temps & Redimensionnement

Nous utilisons une méthode de résolution numérique, introduite dans [48] et développée dans [49], [50] et [51], qui a été initialement conçue pour résoudre des problèmes dont la solution explose en temps fini, et nous l'étendons, dans cette thèse, à la résolution de problèmes dont la solution est globale.
Cette méthode consiste à générer automatiquement des tranches de temps déterminées de façon unique par une condition uniforme de fin de tranches, puis à effectuer un changement de variables, avant de résoudre la suite de problèmes redimensionnés qui en résulte.

a) Condition de fin de tranches

Classiquement, un calcul par tranches de temps commence par le choix d'une grille de temps grossière $\{[T_{n-1}, T_n],\ n \geq 1\}$, souvent régulière.
Notre approche est tout à fait différente et génère automatiquement la grille grossière par le biais d'une condition uniforme de fin de tranche que la solution doit vérifier. Ainsi, on obtient T_n à partir de T_{n-1} dès que la condition de fin de tranche est atteinte par la solution. De proche en proche, la grille grossière est générée de sorte que:

$$\cup_{n\geq 1}[T_{n-1}, T_n] = [0, \infty).$$

Le choix d'une telle condition dépend du comportement de la solution, supposé connu. En général, cette condition est régie par une famille de fonctions continues $\left\{E_n : \mathbb{R}^K \longrightarrow \mathbb{R}\right\}$ qui relient étroitement les valeurs de début et de fin de tranche de la solution. Ces fonctions sont paramétrées par le numéro n de la tranche car elles dépendent souvent de la valeur de début de tranche de la solution $Y_{n-1} = Y\left(T_{n-1}\right)$. Ayant T_{n-1}, elles permettent d'obtenir T_n, de façon unique, lorsque $E_n\left[Y(t)\right] = 0$:

$$(EOS) \qquad \begin{array}{ll} \text{à } t = T_n, & E_n\left[Y\left(T_n\right)\right] = 0, \\ \forall t \in \left(T_{n-1}, T_n\right), & E_n\left[Y\left(t\right)\right] \neq 0. \end{array}$$

La résolution de (S) devient alors équivalente à celle d'une suite infinie de problèmes de tir à valeur initiale:

$$(S_n) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t \geq T_{n-1} \\ Y(T_{n-1}) = Y_{n-1}, & \\ E_n\left[Y(T_n)\right] = 0 \text{ et} & \forall t \in (T_{n-1}, T_n),\ \ E_n\left[Y(t)\right] \neq 0. \end{cases}$$

Tout le long de cette thèse, nous supposons qu'il existe une condition (EOS) de fin de tranche qui génère de façon unique une suite $\{T_n\}$ vérifiant $\cup_{n\geq 1}[T_{n-1}, T_n] = [0, \infty)$.

b) Redimensionnement

Changement de variables:
Sur chacune des tranches $[T_{n-1}, T_n]$, la variable temps t et la solution Y sont redimensionnées en une variable temps s et une solution Z_n, comme suit:

$$(ChVar) \qquad \begin{cases} t = T_{n-1} + \beta_n s, & \beta_n > 0, \\ Y(t) = Y_{n-1} + D_n Z_n(s), & \end{cases}$$

où:

- $Y_{n-1} = Y(T_{n-1}) \in \mathbb{R}^K$
- $D_n = \text{diag}(\alpha_n) \in \mathbb{R}^{K\times K}$ est une matrice diagonale associée à un vecteur $\alpha_n \in \mathbb{R}^K$ qui est defini en fonction de la valeur Y_{n-1} de la solution en début de tranche, par:

$$(\alpha_n) \qquad \alpha_{n,i} = \begin{cases} Y_{n-1,i} & \text{si } Y_{n-1,i} \neq 0 \\ 1 & \text{si } Y_{n-1,i} = 0 \end{cases}$$

- $\beta_n > 0$ est un facteur de redimensionnement de temps qui ne dépend que de la valeur Y_{n-1} de la solution en début de tranche. Ce facteur est choisi de sorte à contrôler la croissance de la solution. Par ailleurs, β_n module les pas de temps redimensionnés constants (en s), générant ainsi des pas de temps réel adaptatifs (en t).

Il est important de noter que le changement de variables $(ChVar)$ initialise le temps et la solution à 0, au début de chaque tranche, et que le choix (α_n) rend la matrice D_n inversible. Par ailleurs, le redimensionnement dépend du choix des paramètres $\{\beta_n\}$: différents choix de β_n mènent à des fonctions Z_n différentes sur une même tranche n. Cependant, on peut remarquer des invariances importantes au niveau des fins de tranches, indépendemment de β_n, obtenues en appliquant le changement de variables $(ChVar)$ à $t = T_n$:

- Le produit $\beta_n s_n$, qui n'est autre que la taille ΔT_n de la $n^{ème}$ tranche, est indépendant du choix de β_n:
$$\forall \beta_n, \quad \Delta T_n = T_n - T_{n-1} = \beta_n s_n,$$
- La valeur $Z_n(s_n)$ de fin de tranche de la solution redimensionnée est indépendante du choix de β_n:
$$\forall \beta_n, \quad Z_n(s_n) = D_n^{-1}(Y_n - Y_{n-1}).$$

Cas de Nonzeroness:
En général, la condition EOS est choisie de sorte à vérifier la condition de nonzeroness suivante:

$$(Nonzeroness) \qquad \forall n, \forall i = 1, 2, \cdots, K, \ \ Y_{n,i} \neq 0.$$

Ceci a pour résultat direct d'avoir $\alpha_n = Y_{n-1}$ $(\forall n)$, ce qui réduit les formules $(ChVar)$ de changement de variables à:

$$(ChVar0) \qquad \begin{cases} t = T_{n-1} + \beta_n s, & \beta_n > 0, \\ Y(t) = D_n \left(\mathbf{1} + Z_n(s)\right), \end{cases}$$

où D_n est simplement la matrice diagonale associé au vecteur Y_{n-1} et $\mathbf{1}$ est le vecteur de $\mathbb{R}^K$ dont toutes les composantes sont égales à 1. Ceci conduit à:

$$\forall n, \ \ \forall i, \quad Z_{n,i}(s) = \frac{Y_i(t) - Y_{n-1,i}}{Y_{n-1,i}},$$

signifiant que la fonction $Z_n(.)$ représente (sous la condition de nonzeroness) la variation relative de $Y(.)$ par rapport à Y_{n-1}, au sein de la $n^{ème}$ tranche $[T_{n-1}, T_n]$ (voir figure 1.1).

Redimensionnement de la condition EOS:
Soit $Z_n(s_n)$ la valeur de fin de tranche de la solution redimensionnée, à $s = s_n$.
La condition (EOS) de fin de tranche, régie par les fonctions $\left\{E_n : \mathbb{R}^K \longrightarrow \mathbb{R}\right\}$, devient, après le changement de variables $(ChVar)$:

$$\begin{array}{ll} \text{à } s = s_n, & E_n \left[Y_{n-1} + D_n Z_n(s_n)\right] = 0, \\ \forall s \in (0, s_n), & E_n \left[Y_{n-1} + D_n Z_n(s)\right] \neq 0. \end{array}$$

Ceci définit une nouvelle famille $\{H_n\}$ de fonctions continues, paramétrées aussi par les valeurs $\{Y_{n-1}\}$ de la solution en début de tranches, permettant d'obtenir s_n, pour tout n, par la condition EOS redimensionnée:

$$\begin{array}{ll} \text{à } s = s_n, & H_n \left[Z_n(s_n)\right] = 0, \\ \forall s \in (0, s_n), & H_n \left[Z_n(s)\right] \neq 0. \end{array}$$

Systèmes redimensionnés équivalents:
Le choix (EOS) de la condition de fin de tranches, avec le changement de variables $(ChVar)$, rendent la résolution du problème (S) équivalente à la résolution d'une suite de problèmes de tirs redimensionnés et à valeur initiale. Il faut ainsi trouver le temps redimensionné s_n et la solution redimensionnée $Z_n : [0, s_n] \to \mathbb{R}^K$, sur chaque $n^{ème}$ tranche correspondant à $[T_{n-1}, T_n]$, satisfaisant:

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, \\ H\left[Z_n(s)\right] \neq 0, \ \forall s < s_n \ \text{ et } \ H\left[Z_n(s_n)\right] = 0, \end{cases}$$

où les fonctions $\{E_n\}$ régissant la condition EOS sont supposées pouvoir être redimensionnées en une fonction invariante H.

c) Propriétés de similarité:

Le calcul en tranches de temps avec redimensionnement, décrit ci-dessus, a été utilisé dans [48], [49], [50] et [51], et a donné des résultats de grande précision dans le cas de problème où la solution explose en temps fini. Tout porte à croire que cette méthode peut être étendue aux problèmes à temps d'existence global, en vue (i) de contrôler la croissance et la précision de la solution dans le cas de problèmes raides et (ii) de construire un schéma parallèle en temps pour la résolution de (S). C'est ce dernier objectif qui est visé dans

cette thèse.
Mais une telle extension commence par la définition de quelques propriétés de similarité que doivent vérifier les systèmes redimensionnés ($S'n$):

1. *Invariance:*
   Les systèmes redimensionnés (S'_n) sont dits invariants lorsque les fonctions $\{H_n\}$ régissant la condition EOS redimensionnée sont invariantes, et les paramètres $\{\beta_n\}$ de redimensionnement de la variable temps sont tels que:

   $$\forall n, \quad G_n(.) = G_1(.).$$

   Puisque toutes les fonctions G_n sont égales à G_1, alors les systèmes redimensionnés (S'_n) sont tous équivalents à (S'_1), impliquant:

   $$\forall n, \quad Z_n(.) = Z_1(.) \qquad \text{et} \qquad s_n = s_1.$$

   L'invariance est le cas idéal de similarité: résoudre le problème sur une seule tranche de temps suffit alors pour avoir la solution sur toutes les tranches, par un simple changement de variables.

2. *Similarité asymptotique*:
   Les systèmes redimensionnés(S'_n) sont dits asymptotiquement similaires à un problème limite lorsque les fonctions $\{H_n\}$ régissant la condition EOS redimensionnée sont invariantes et les paramètres $\{\beta_n\}$ de redimensionnement de la variable temps sont tels que la suite de fonctions $\{G_n\}$ converge uniformément à une fonction G_L, i.e.:

   $$\forall \rho > 0, \quad \lim_{n\to\infty} \left[\max_{W \in B_\rho} \|G_n(W) - G_L(W)\|_\infty \right] = 0,$$

   où $B_\rho = \left\{ W \in \mathbb{R}^K, \ \|W\|_\infty \leq \rho \right\}$.
   La fonction $G_L : \mathbb{R}^K \longrightarrow \mathbb{R}^K$ definit alors un "problème limite" dans lequel il faut trouver le temps redimensionné s_L et la solution redimensionnée $Z_L : [0, s_L] \to \mathbb{R}^K$, vérifiant:

   $$(S_L) \quad \begin{cases} \frac{dZ_L}{ds} = G_L(Z_L), & 0 < s \leq s_L, \\ Z_L(0) = 0, & \\ H\left[Z_L(s)\right] \neq 0, \ \forall s < s_L \ \text{ et } \ H\left[Z_L(s_L)\right] = 0, & \end{cases}$$

   Sous certaines conditions, cette propriété entraîne la convergence des valeurs $\{s_n\}$ et $\{Z_n(s_n)\}$ de fin de tranches vers s_L et $Z_L(s_L)$.

3. *Similarité faible*:
   On dit que les systèmes redimensionnés (S'_n) présentent une similarité faible (i.e. numérique) sur n_r tranches consécutives, commençant à la tranche n_0, lorsque les fonctions $\{H_n\}$ régissant la condition EOS redimensionnée sont invariantes et que:

   $$\exists \epsilon << 1, \ \forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \quad \|Z_n(s_n) - Z_{n-1}(s_{n-1})\|_\infty < \epsilon.$$

   Nous avons remarqué dans nos essais numériques, et dans le cas de choix de conditions EOS bien appropriées au comportement de la solution, qu'une similarité faible vérifiée sur n_r tranches consécutives, commençant à la tranche n_0, continue à être vérifiée bien au-delà.
   Il est clair que l'invariance et la similarité asymptotique sont des cas particuliers (et bien plus forts) de la similarité faible.

4. *Similarité uniforme*:
   Les problèmes à valeur initiale:

$$(IVP_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

   sont dits uniformément similaires lorsque les paramètres $\{\beta_n\}$ de redimensionnement de la variable temps sont tels que:

$$\forall n \geq 1,\ \forall S > 0,\ \exists C, C' > 0, \quad ||W||_\infty \leq S \implies \begin{cases} ||G_n(W)||_\infty \leq C, \\ ||J_{G_n}(W)||_\infty \leq C', \end{cases}$$

   où $J_{G_n}(.) : \mathbb{R}^K \to \mathbb{R}^K \times \mathbb{R}^K$ est le Jacobien de G_n. En d'autres termes, si la solution Z_n est bornée, alors $G_n(Z_n)$ et $J_{G_n}(Z_n)$ sont bornés uniformément (i.e. indépendemment de n). Cette propriété de similarité uniforme permet ainsi de contrôler la raideur d'un problème.

De ces propriétés de similarité, on retient surtout que dans les cas d'invariance, de similarité asymptotique et de similarité faible, les valeurs $Z_n(s_n)$ de fin de tranche de la solution redimensionnée sont invariantes ou presque-invariantes. On montre alors, dans ce qui suit, comment ces 3 propriétés peuvent conduire à la construction d'un schéma de calcul parallèle en temps.

d) Exemples de choix de conditions EOS

Nous donnons des choix possibles de telles conditions, dans les cas de solutions explosives et oscillatoires qui seront expérimentés dans les Chapitres 5 à 7. Puis nous proposons des extensions à d'autres cas, cependant non expérimentés.

1. *Problèmes explosifs :* $\lim_{t\to\infty} \|Y(t)\|_\infty = \infty$
   Dans ce cas, les conditions EOS proposées visent à contrôler la croissance de la solution redimensionnée.

   - *Cas général :*
     Les fonctions régissant la condition (EOS) sont définies, pour un seuil choisi S et pour tout n, par:

$$\forall W \in \mathbb{R}^K, \quad E_n(W) = \|W - Y_{n-1}\|_\infty - S\,\|Y_{n-1}\|_\infty .$$

     La condition (EOS) s'écrit alors explicitement:

$$\begin{array}{c} \text{à } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \|Y\left(T_n\right) - Y_{n-1}\|_\infty = S\,\|Y_{n-1}\|_\infty , \\ \text{avec} \quad \|(Y(t) - Y_{n-1})\|_\infty < S\,\|Y_{n-1}\|_\infty , \quad \text{si } T_{n-1} < t < T_n. \end{array}$$

     La continuité et la nature explosive de la solution garantissent d'atteindre la condition EOS, de façon unique, pour tout n.
     L'expression de la condition redimensionnée est:

$$\begin{array}{c} \text{à } s = s_n, \quad H_n\left[Z_n(s_n)\right] = 0 \iff \|D_n Z_n(s_n)\|_\infty = S\,\|Y_{n-1}\|_\infty , \\ \text{avec} \quad \|D_n Z_n(s)\|_\infty < S\,\|Y_{n-1}\|_\infty , \quad \text{si } 0 < s < s_n. \end{array}$$

     Clairement, les fonctions $\{H_n\}$ régissant cette condition et définies par:

$$\forall W \in \mathbb{R}^K,\ H_n(W) = \|D_n W\|_\infty - S\,\|Y_{n-1}\|_\infty ,$$

ne sont pas invariantes.

- *Cas de nonzeroness:* dans le cas où la condition ($Nonzeroness$) est vérifiée, un autre choix possible serait de définir les fonctions régissant la condition (EOS), pour un seuil choisi S et pour tout n, par:

$$E_n(W) = \left\| D_n^{-1} \left(W - Y_{n-1} \right) \right\|_\infty - S,$$

où D_n est ici la matrice diagonale associé au vecteur Y_{n-1}.
L'effet de la condition (EOS) qui en résulte est alors d'empêcher la valeur absolue de la variation relative de chacune des composantes de $Y(t)$ de dépasser le seuil S:

$$\begin{array}{c} \text{à } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \left\| D_n^{-1} \left(Y(T_n) - Y_{n-1} \right) \right\|_\infty = S, \\ \text{avec} \quad \left\| D_n^{-1} \left(Y(t) - Y_{n-1} \right) \right\|_\infty < S, \quad \text{si } \ T_{n-1} < t < T_n. \end{array}$$

Là aussi, la continuité et la nature explosive de la solution garantissent d'atteindre la condition EOS, de façon unique, pour tout n.
Par contre l'expression de la condition redimensionnée est ici:

$$\begin{array}{c} \text{à } s = s_n, \quad H\left[Z_n(s_n)\right] = 0 \iff \left\| Z_n(s_n) \right\|_\infty = S, \\ \text{avec} \quad \left\| Z_n(s) \right\|_\infty < S, \quad \text{si } 0 < s < s_n. \end{array}$$

montrant clairement l'invariance des fonctions $\{H_n\}$ régissant cette condition et définies par:

$$\forall W \in \mathbb{R}^K,\ H(W) = \left\| W \right\|_\infty - S.$$

2. *Problèmes à solution oscillatoire :*
On considère ici les problèmes dont la solution est oscillatoire, dans ce sens qu'il existe un plan P de $\mathbb{R}^K$, de dimension 2, dans lequel la projection de la solution $Y(t)$ tourne indéfiniment autour d'un point fixe ω. On choisit alors de terminer une tranche de calcul à chaque fois que la solution effectue une rotation complète (ou presque complète) dans ce plan. Soit (i_1, i_2) ce plan et on suppose, sans perte de généralité, que la rotation a lieu autour de l'origine ω.

   - *Choix général :*
   Soit $M(t) = (Y_{i_1}(t), Y_{i_2}(t))$ la projection de la solution $Y(t)$ sur le plan P, à tout instant t. La projection de la valeur Y_{n-1} de la solution, en début de tranche, est alors le point : $M_{n-1} = (Y_{i_1}(T_{n-1}), Y_{i_2}(T_{n-1})) = (Y_{n-1,i_1}, Y_{n-1,i_2})$.
   En tournant autour de ω, le point $M(t)$ definit dans le plan P, un angle de rotation qui croît de façon monotone (voir figure 2.2) :

$$\theta_n\left[Y(t)\right] = \left| \left(\omega\vec{M}_{n-1}, \omega\vec{M}(t) \right) \right|.$$

   On définit alors des fonctions $\{E_n\}$ régissant la condition (EOS), pour tout n et en fonction des valeurs $\{Y_{n-1}\}$ de début de tranches, par:

$$\forall Y(t) \in \mathbb{R}^k, \quad E_n\left[Y(t)\right] = \theta_n\left[Y(t)\right] - 2\pi,$$

   où $\theta_n(W)$ est calculé en utilisant la procédure donnée en définition 7.
   La $n^{ème}$ tranche est ainsi terminée lorsque la solution effectue une rotation

complète dans le plan P, c'est-à-dire lorsque :

$$\begin{array}{c} \text{a } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0, \quad \Longleftrightarrow \quad \theta_n\left[Y(T_n)\right] = 2\pi, \\ \text{avec} \quad \theta_n\left[Y(t)\right] < 2\pi,\ \text{ si } T_{n-1} \leq t < T_n. \end{array}$$

Il en résulte que l'angle polaire φ_n, à la fin de chaque $n^{ème}$ tranche, est égal à l'angle polaire initial φ_0 qui correspond à la valeur initiale Y_0 de la solution, rendant ainsi les fonctions $\{\theta_n\}$ et $\{E_n\}$ indépendantes de n.
Il est à noter que la nature oscillatoire de la solution garantit d'atteindre cette condition EOS, de façon unique, pour tout n (voir figure 2.3).
On montre, dans le cas où la condition ($Nonzeroness$) est vérifiée, que les fonctions $\{H_n\}$, déduites des fonctions $\{E_n\}$ après redimensionnement, sont invariantes et définies pour tout n par :

$$\forall Z_n(s) \in \mathbb{R}^k,\ H\left[Z_n(s)\right] = \tilde{\theta}\left[Z_n(s)\right] - 2\pi,$$

où $\tilde{\theta}\left(.\right)$ est l'expression redimensionnée de la fonction $\theta\left(.\right)$ (ayant $\forall n$, $\theta_n = \theta$).
La $n^{ème}$ tranche de temps est ainsi terminée lorsque :

$$\begin{array}{c} \text{à } s = s_n, \qquad H\left[Z_n(s_n)\right] = 0 \quad \Longleftrightarrow \quad \tilde{\theta}\left[Z_n(s_n)\right] = 2\pi, \\ \text{avec} \quad \tilde{\theta}\left[Z_n(s)\right] < 2\pi, \qquad \text{si } 0 < s < s_n. \end{array}$$

- *Autres choix possibles :*
  - Dans la condition EOS décrite ci-dessus, une tranche de temps est terminée à chaque fois que la trajectoire de $M(t)$, dans le plan de projection P, coupe la demi-droite faisant un angle φ_0 avec l'axe (ω, i_1).
    Une variante de cette condition pourrait être de terminer une tranche de temps à chaque fois que la trajectoire de $M(t)$, dans le plan de projection P, coupe, dans le quadrant qui contient la position initiale M_0, la représentation graphique C d'une fonction continue $\Phi(.)$, définie sur l'intervalle de $\mathbb{R}$ correspondant à ce quadrant. Cette variante a été utilisée dans le Chaptitre 5, dans une application à un problème de membrane, et a eu l'avantage de faire apparaître un cas d'invariance.
  - Dans le cas où la solution traverse le plan P à chaque rotation, on pourrait choisir de terminer une tranche de temps à chaque fois que la solution traverse le plan de projection après avoir effectué une rotation complète.
    Cette variante a été utilisée dans le Chaptitre 7, dans une application à un problème de trajectoire de satellite dans un mouvement non keplerien, connue pour être une ellipse osculatrice traversant régulièrement le plan Keplerien correspondant aux condition initiales. Elle a eu pour résultat de produire une propriété des ratios plus importante.
  - Dans le cas où le problème considéré a une solution à la fois explosive et oscillatoire, et que l'objectif est de contrôler la raideur du problème seulement (sans se soucier de l'existence de propriété de ratios, donc sans objectif de parallélisation), il serait bon de choisir des conditions EOS similaires à celles proposées pour les problèmes explosifs. Ceci a été fait dans [51] pour résoudre (séquentiellement) un problème de membrane dont la solution est oscillatoire et explosive en temps fini. Une propriété de similarité uniforme a été obtenue; le temps d'explosion et la solution ont été obtenus de façon très précise.

3. *Extension à d'autres cas de comportement de la solution:*
   Bien que non expérimentés numériquement, et dans un souci de sonder l'applicabilité de la méthode de calcul en tranches, des conditions EOS sont proposées pour des problèmes dont la solution a l'un des comportement suivants :

   - extinctif : $\lim_{t\to\infty} \|Y(t)\|_\infty = 0$,
   - borné avec un point d'attraction à l'infini : $\lim_{t\to\infty} Y(t) = l \neq 0 \in \mathbb{R}^K$.

e) Cas particulier d'une fonction F algébrique

On considère le cas particulier où la fonction F, définissant le problème (S), a une forme algébrique donnée, pour chaque $i^{ème}$ composante, par :

$$(F(Y))_i = \sum_j a_{ij} (Y_i)^{k_{ij}} (Y_j)^{l_{ij}},$$

avec $a_{ij} \in \mathbb{R}$, et k_{ij}, l_{ij} étant des coefficients entiers tels que : $\forall i, \forall j, \quad k_{ij} \geq 1, \ l_{ij} \geq 1$ (F satisfaisant alors la propriété de Lipschitz).
On démontre, dans ce cas et pour une suite positive $\{\rho_n\}$ donnée, qu'il existe une suite $\{(T_n, Y_n)\}_{n=1,2,\dots} \subset (0,\infty)\times\mathbb{R}^K$, où $Y_n = Y(T_n)$, déduite de $(T_0 = 0, Y_0)$ et $\{\rho_n\}$, telle que $\{T_n\}$ est croissante et le problème (S) a une solution unique $Y : [0, T_m] \to \mathbb{R}^K$ vérifiant : $\forall n \in 1, 2, \cdots, m, \ \forall t \in [T_{n-1}, T_n], \quad \|Y(t) - Y_{n-1}\|_\infty \leq \rho_n$.
On montre ensuite comment, de cette étude d'existence et d'unicité de la solution, on peut déduire l'existence d'une suite croissante $\{T'_n\}$ et d'une suite positive $\{\rho'_n\}$, avec $\forall n, \ \rho'_n \leq \rho_n$ et $T'_n \leq T_n$, qui rendent la résolution du problème (S) équivalente à celle d'une suite de problèmes (S_n) de tirs, à valeur initiale, dans lesquels la condition EOS est donnée par la fonction:

$$E_n\left(Y(t)\right) = \|Y(t) - Y_{n-1}\|_\infty - \rho'_n,$$

et s'écrit :

$$\begin{array}{c} \text{à } t = T_n, \ E_n\left[Y\left(T_n\right)\right] = 0 \iff \|Y\left(T_n\right) - Y_{n-1}\|_\infty = \rho'_n, \\ \text{avec} \quad \|(Y(t) - Y_{n-1})\|_\infty < \rho'_n, \quad \text{si} \ \ T_{n-1} < t < T_n. \end{array}$$

L'intérêt de cette étude est de prouver l'existence d'une condition EOS générant une grille grossière, *indépendemment du comportement de la solution* !

Dans ce même cas de fonction algébrique, nous montrons aussi l'existence d'un choix possible de paramètres $\{\beta_n\}$ menant à la similarité uniforme des systèmes.
Nous montrons ensuite comment se traduisent les propriétés d'invariance et de similarité asymptotique, dans ce cas algébrique.

Un cas algébrique particulier est enfin considéré : le cas linéaire, obtenu lorsque $\forall i, \ \forall j \ \ k_{ij} = 0$ et $l_{ij} = 1$. Nous prouvons ici, l'existence d'une suite $\{\beta_n\}$ qui, sous certaines conditions, mène à une similarité uniforme et asymptotique.

II- Schéma de Calcul Parallèle en temps

Dans cette partie, nous supposons la condition $(Nonzeroness)$ satisfaite et nous définissons des propriétés de ratios correspondants aux propriétés d'invariance et quasi-invariance des systèmes redimensionnés. Nous montrons ensuite comment, après une étude séquentielle

préliminaire sur un certain nombre n_s de tranches, on peut construire un schéma parallèle de calcul, basé sur la propriété des ratios. Ce schéma est concrétisé par l'algorithme RaPTI: "Ratio-based Parallel Time Integration".

a) Propriété des ratios

Dans le cas où la condition (*Nonzeroness*) est satisfaite, les valeurs $Z_n(s_n)$ de fin de tranche de la solution redimensionnée, sur la $n^{ème}$ tranche, relient directement les valeurs consécutives Y_{n-1} et Y_n de début de tranches de la solution:

$$Y_n = D_n \left(\mathbf{1} + Z_n(s_n)\right),$$

où D_n est la matrice diagonale associé au vecteur Y_{n-1} et $\mathbf{1}$ est le vecteur de $\mathbb{R}^K$ dont toutes les composantes sont égales à 1.
En posant:

$$R_n = \mathbf{1} + Z_n(s_n),$$

et D_{R_n} la matrice diagonale associée au vecteur R_n, la relation précédente devient $Y_n = D_n R_n$, qui n'est autre qu' une multiplication par composante des vecteurs R_n et Y_{n-1}, et s'écrit de façon équivalente :

$$Y_n = D_{R_n} Y_{n-1}.$$

Ceci montre comment R_n peut être considéré comme un "ratio-vecteur" reliant les valeurs consécutives Y_{n-1} et Y_n de début de tranches de la solution.
On peut alors déduire que, dans les cas d'invariance, de similarité asymptotique et de similarité faible, les ratio-vecteurs $\{R_n = \mathbf{1} + Z_n(s_n)\}$ sont invariants ou presque-invariants. On dit alors que la suite $\{Y_n\}$ des valeurs de début de tranches de la solution a une "*propriété des ratios*" (parfaite, asymptotique ou faible, selon le cas de similarité).
Cette quasi-stabilisation des ratios est le principe de départ de notre procédure de prédiction: l'idée de base en est de résoudre le problème séquentiellement jusqu'à quasi-stabilisation des ratios, puis de prédire les valeurs suivantes de Y_n en utilisant les derniers ratios presque stabilisés. D'où la terminologie: "*ratio-based prediction procedure*".

b) Cas particulier d'invariance

Dans le cas d'invariance des systèmes redimensionnés, on obtient des ratios invariants :

$$\forall n > 0,\ R_n \ = \ R_1.$$

Grâce à cette propriété parfaite des ratios, la résolution séquentielle du problème sur la première tranche permet d'obtenir des prédictions exactes, au début de toutes les autres tranches :

$$\forall n, \quad Y_n = \left(D_{R_1}\right)^n Y_0 = \left(I + D_{Z_1(s_1)}\right)^n Y_0.$$

Ceci conduirait à une parallélisation parfaite des calculs, sans aucune communication et en une seule itération.
Cependant, le parallélisme devient tout à fait inutile dans ce cas. En effet, résoudre le problème sur la première tranche le résoud sur toutes les tranches, en utilisant simplement, et de façon récurrente, le changement de variables suivant :

$$\begin{cases} t = T_{n-1} + \beta_n s, & \beta_n > 0 \\ Y(t) = D_n \left(\mathbf{1} + Z_1(s)\right), \end{cases}$$

Ceci est fait au moyen d'un algorithme RaTI, "Ratio-based Time Integration", propre aux cas invariants (voir section 4.2).

c) Etude préliminaire

Dans le cas plus général de quasi-invariance (i.e. de similarité asymptotique ou faible), et avant d'appliquer l'algorithme RaPTI ("Ratio-based Parallel Time Integration"), une étude séquentielle préliminaire doit être faite. Elle a trois objectifs :

1. *Justifier l'applicabilité de RaPTI par la détection d'une propriété des ratios* : Ceci est fait par le biais d'une procédure "DETECT_RATIO_PROPERTY_UP_TO_ϵ", détaillée en annexe 1, qui consiste à résoudre les systèmes (S'_n) séquentiellement, sur un certain nombre de tranches. Une propriété des ratios, à une tolérance ϵ, est dite numériquement détectée à la tranche n_0 et sur n_r tranches consécutives, et donc atteinte à la tranche $n_s = n_0 + n_r$, si:

$$\forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \quad \|R_n - R_{n-1}\|_\infty < \epsilon.$$

   Un garde-fou arrête cette procédure si une telle propriété n'est pas atteinte au bout d'un nombre acceptable n_{stop} de tranches: la méthode n'est pas applicable alors.

2. *Extraire de la suite de ratios $\{R_n\}_{n \leq n_s}$, un modèle mathématique susceptible de donner de bonnes estimations des ratios suivants $\{R_n\}_{n > n_s}$.*
   Dans le cas où une propriété de ratios est atteinte (i.e. les ratios sont quasi-stabilisés), l'idée de base de notre procédure de prédiction est de supposer que les ratios suivants ne varient plus que faiblement, d'une tranche à l'autre, en suivant une tendance qui peut être approximée par un modèle mathématique. Plusieurs modèles doivent être considérés : celui qui extrapole le mieux sera choisi comme modèle de prédiction.
   Le modèle le plus simple, mais le moins efficace, table sur la quasi-stabilisation des ratios et utilise le dernier ratio R_{n_s} calculé pour approximer les ratios suivants :

$$\forall n \geq n_s, \qquad R^p_n = R_{n_s}.$$

   La suite $\{R^p_n\}_{n > n_s}$ est donc supposée constante (comme dans le cas d'invariance) et approxime chaque composante des ratios suivants par une suite géométrique :

$$\forall n \geq n_s, \quad \forall i \in \{1, 2, \cdots, k\}, \qquad Y^p_{n,i} = (R_{n_s,i})^{n-n_s}\, Y^e_{n_s,i}.$$

   où $Y^e_{n_s}$ est la valeur de la solution à $t = T_{n_s}$ séquentiellement calculée.

   D'autres modèles, plus sophistiqués, sont testés : ils se basent tous sur des techniques statistiques qui tentent d'ajuster les ratios séquentiellement calculés par des fonctions mathématiques. Chaque modèle considéré est caractérisé par sa forme mathématique (polynomiale, exponentielle, logarithmique, ...) et par le nombre de ratios qu'il utilise pour l'ajustement. Dans toutes nos expériences numériques, il s'est avéré que l'usage d'un modèle polynomial avec un nombre minimal de ratios (les derniers à avoir été séquentiellement calculés) donnait les meilleures prédictions:

$$R_{fit}(n) = a_d n^d + \cdots + a_1 n + a_0,$$

   où d est le degré du polynôme et $a_0, \cdots, a_d$ en sont les paramètres.

Dans le cas de similarité asymptotique, le modèle considéré devrait avoir une limite égale au ratio limite qui peut être calculé. Une façon de garantir cette convergence serait de chercher un ajustement de $[n\,(R_n - R_L)]$ sous la forme d'un polynôme en $\frac{1}{n}$, de degré d, impliquant :
$n\,(R_n - R_L) \approx n\,(R_{fit}(n) - R_L) = a_d\left(\frac{1}{n}\right)^d + \cdots + a_1\left(\frac{1}{n}\right) + a_0$.
On en déduit :

$$R_n \approx R_{fit}(n) = R_L + \frac{1}{n}\left[a_d\left(\frac{1}{n}\right)^d + \cdots + a_1\left(\frac{1}{n}\right) + a_0\right],$$

et par suite :

$$\lim_{n\to\infty} R_{fit}(n) = R_L.$$

Il est à noter qu'un ajustement de degré d (en $1/n$) pour $[n\,(R_n - R_L)]$, correspond à un ajustement polynomial de degré $(d+1)$ pour R_n.

Enfin, on fait subir à tous les modèles considérés un test d'extrapolation. Ce test consiste à résoudre le problème séquentiellement sur n_{add} tranches supplémentaires, donnant ainsi un ensemble de validation $\{R_{n_s+1}, \cdots, R_{n_s+n_{add}}\}$ formé de n_{add} ratios exacts supplémentaires. Chaque $j^{ème}$ modèle étudié est alors utilisé pour approximer les n_{add} ratios supplémentaires puis les comparer à l'ensemble de validation et en déduire son erreur maximale :

$$error_j = \max_{n_s<n\leq n_s+n_{add}}\left[\max_{i\in\{1,2,\cdots,K\}}\left|R_{fit_j,i}(n) - R_{n,i}\right|\right].$$

Le modèle donnant la plus petite erreur est alors choisi. Ceci est fait au moyen de la procédure " FIND_RATIOS_MODEL ", donnée en annexe 1.
Cependant, ce test d'extrapolation ne peut pas donner la portée du modèle, c'est-à-dire qu'on ne peut pas savoir d'avance jusqu'où le modèle continuera à donner de bonnes approximations. Une telle portée sera déterminée par le nombre de tranches qui convergent à la première itération.

3. *Déterminer, de façon approximative, le nombre total N de tranches* sur lequel le problème doit être résolu, de sorte à couvrir l'intervalle de calcul $[0, T]$ demandé : cette approximation est nécessaire puisque la taille des tranches de temps n'est pas initialement choisie, mais obtenue au fur et à mesure de la résolution du problème.
Une procédure assez élaborée, "PREDICT_AND_ESTIMATE_N", est proposée en annexe 1 et consiste à estimer ce nombre durant la phase de prédiction, en prédisant aussi les temps $\{T_n\}$.
Cependant, on peut souvent donner une approximation moins précise mais plus rapide de N, quitte à surestimer N puis arrêter les calculs dès que T est atteint.
Dans le cadre de cette thèse, les problèmes étudiés sont :
- soit explosifs (Chapitres 5 et 6), N est alors limité par la capacité de la machine,
- soit quasi-périodiques (Chapitre 7), N est alors estimé en approximant la taille des tranches par celle du problème périodique correspondant.

Après cette étude préliminaire, on devrait être à même de prédire les valeurs $\{Y_n\}_{n>n_s}$ de début de tranches de la solution et d'envisager le calcul parallèle des tranches $\{n_s+1, n_s+2, \cdots, N\}$.
Il est à noter, comme nos expériences numériques l'indiquent, que le modèle mathématique

trouvé, lors de l'étude préliminaire, continue à être bien approprié à toutes les itérations, après avoir cependant mis à jour les paramètres du modèle, conformément aux derniers ratios.

d) Résolution parallèle des systèmes redimensionnés

Chaque $n^{ème}$ système redimensionné ($n > n_s$) est résolu, en commençant avec une valeur initiale Y^p_{n-1} prédite. Ceci a pour effet de remplacer les valeurs β_n, D_n et G_n (qui dépendent toutes de Y_{n-1}) par leurs valeurs prédites β^p_n, D^p_n et G^p_n, respectivement. Il faudrait alors trouver le temps s^c_n et la solution $Z^c_n : [0, s_n] \to \mathbb{R}^K$ redimensionnés, satisfaisant :

$$\begin{cases} \frac{dZ^c_n}{ds} = G^p_n(Z^c_n), & 0 < s \leq s^c_n, \\ Z^c_n(0) = 0, & \\ H\left[Z^c_n(s)\right] \neq 0, \ \forall s < s^c_n \ \text{ et } \ H\left[Z^c_n(s^c_n)\right] = 0 & \end{cases}$$

où $G^p_n(Z^c_n) = \beta^p_n (D^p_n)^{-1} F\big(Y^p_{n-1} + D^p_n Z^c_n\big)$.
La résolution de ce système conduit aux valeurs redimensionnées corrigées s^c_n and $Z^c_n\left(s^c_n\right)$ de fin de tranche, desquelles sont déduites les valeurs corrigées Y^c_n et T^c_n de fin de tranche:

$$Y^c_n = Y^p_{n-1} + D^p_n Z^c_n(s^c_n),$$

$$T^c_n = T^c_{n-1} + \beta^p_n s^c_n.$$

Il faut remarquer que les temps $\{T_n\}$ ne sont pas prédits : ils sont obtenus au fur et à mesure de la progression des calculs.

Nos essais numériques ont montré que la taille redimensionée $\{s^c_n\}$, ainsi que la solution redimensionnée $Z_n(.)$ de la $n^{ème}$ tranche, sont très proches de leurs valeurs exactes, tant que les valeurs prédites sont suffisamment bonnes.
En supposant ceci prouvé (chose qu'on n'a pas encore réussi à faire), on montre que Y^c_n et T^c_n convergent alors vers leurs valeurs exactes, tant que la prédiction Y^p_{n-1} est assez bonne et que T^c_{n-1} est validé. Ceci conduit à un test de convergence qui porte sur les sauts $\|Y^p_n - Y^c_n\|_\infty$: tant que ces sauts sont inférieurs à une tolérance donnée, les prédictions continuent à pouvoir être efficacement utilisées pour calculer la tranche suivante.

e) Algorithme RaPTI

L'algorithme RaPTI est un schéma itératif de calcul qui permet de résoudre en parallèle les systèmes redimensionnés (S'_n).
Les premières n_s tranches sont résolues séquentiellement par tous les processeurs puis les tranches suivantes ($n > n_s$) sont *statiquement réparties sur les n_p processeurs disponibles, de façon cyclique.* Le $j^{ème}$ processeur se voit attribuer toutes les tranches dont le numéro n vérifie :

$$n = n_s + j + l n_p,$$

où $l \in \{0, 1, \cdots \left\lfloor \frac{N - n_s - j}{n_p} \right\rfloor\}$, signifiant que $(n - n_s)$ est congruent à j modulo n_p.
Cette distribution est schématisée par la figure 4.4.
Outre sa simplicité, l'intérêt d'une telle distribution cyclique réside dans le fait qu'elle facilite la synchronisation des calculs parallèles : à chaque itération, les divers processeurs arrêteront le calcul parallèle sur des tranches consécutives (lorsque les prédictions cessent d'être bonnes), donc presque en même temps.

Par ailleurs, tous les calculs séquentiels sont dupliqués sur les processeurs, de sorte à minimiser les communications et les temps d'attente. Une seule étape de communication doit se faire après chaque itération, comme l'indique la figure 4.5, en vue de passer les informations nécessaires à la mise à jour des prédictions.
Ainsi, après avoir mené à bien l'étude séquentielle préliminaire, l'implémentation parallèle de l'algorithme RaPTI se fait selon une distribution cyclique des tranches et en utilisant l'approche duplicative décrite ci-dessus. Le schéma est formée de deux étapes principales, comme détaillé ci-dessous.

Algorithme RaPTI

- **Etape 1 : Phase d'initialisation**
  - *Atteinte de la propriété des ratios* :
    Calculs séquentiels, dupliqués sur tous les processeurs, aboutissant à un nombre n_s de tranches au bout desquelles une propriété des ratios est atteinte, à une tolérance $\epsilon_{tol}^{n_s}$, et sur lesquels le problème est complètement résolu par tous les processeurs.
  - *Initialisation* du numéro d'iteration à $k = 1$, et du nombre de tranches ayant convergé avant la $1ère$ itération à $n_s^{(k-1)} = n_s$.
- **Etape 2 : Procédure itérative**
  - **Etape 2.1 : Prédiction**
    - *Mise à jour des paramètres du modèle mathématique* :
      Calculs séquentiels, dupliqués sur tous les processeurs.
    - *Prédictions* : Calculs séquentiels, dupliqués sur tous les processeurs et utilisant le modèle mathématique mis à jour, déterminant ainsi les prédictions sur toutes les tranches restantes ($n_s^k < n \leq N$).
  - **Etape 2.2 : Correction**
    Calculs parallèles : Chaque processeur résoud les tranches qui lui sont attribuées, tant qu'aucune tranche n'a divergé.
  - **Etape 2.3 : Communication**
    Etape de communication qui a pour but de déterminer le numéro $n_s^{(k)}$ de la dernière tranche ayant convergé à la $k^{ème}$ itération, ainsi que les temps $\{T_n^c\}_{n_s^{(k-1)} < n \leq n_s^{(k)}}$ calculés.
    Si $T^c_{n_s^{(k)}} < T$, alors les dernières n_{prec} valeurs de Y_n sont aussi communiquées, en vue de mettre à jour les prédictions.
  - **Etape 2.4 :**
    Si $T^c_{n_s^{(k)}} < T$, alors $k = k+1$ et l'étape 2 est répétée jusquà ce que $T^c_{n_s^{(k)}} \geq T$.

Conclusion
Chaque processeur affiche la solution $Y(t)$ sur les tranches de temps qu'il a calculées.

Etape de prédiction :
Théoriquement, la procédure FIND_RATIOS_MODEL, utilisée en phase préliminaire pour déterminer le modèle mathématique à utiliser dans les prédictions, devrait être répétée à chaque itération. Cependant, nos essais numériques montrent que, pour un problème donné, la même forme mathématique et le même nombre n_{prec} de ratios précédents continuent à approximer le mieux les ratios suivants. Il suffit alors, en début de chaque itération, de mettre à jour les paramètres de ce modèle, en les recalculant de sorte à ajuster (au sens des moindres carrés) les ratios des dernières tranches ayant convergé à la fin de l'itération précédente. Ceci est fait par le biais de la procédure "GET_MODEL_PARAMETERS", donnée en annexe 1, qui aboutit au modèle $R_{fit}^{(k)}$ à utiliser à la $k^{ème}$ itération.
La procédure "PREDICT", donnée en annexe 1 aussi, utilise le nouveau modèle $R_{fit}^{(k)}$ pour approximer les ratios suivants $\{R_n^p\}_{n>n_s^{(k)}}$ qui sont nécéssaires aux prédictions de la $k^{ème}$ itération :

$$\forall n > n_s^{(k)}, \qquad R_n^{p^{(k)}} = R_{fit}^{(k)}(n).$$

Les valeurs de débuts de tranches $\{Y_n\}_{n>n_s^{(k)}}$ de la solution sont alors prédites de façon récurrente, en utilisant les ratios prédits et la dernière valeur caculée $Y^c_{n_s^{(k)}}$ de la solution :

$$\forall n > n_s^{(k)}, \qquad Y_n^{p^{(k)}} = D_{R_n^{p^{(k)}}} \cdots D_{R_{n_s+1}^{p^{(k)}}} Y^c_{n_s^{(k)}}.$$

Dans le cas où l'on veut avoir, en même temps, une estimation assez pointue du nombre N total de tranches à calculer, la procédure ci-dessus est remplacée par la procédure PREDICT_AND_ESTIMATE_N.

Etape de correction :
Les corrections sont faites en parallèle, après chaque prédiction.
A la $k^{ème}$ itération, et pour $n > n_s^{(k-1)}$, chaque $j^{ème}$ processeur commence à résoudre ses tranches (qui lui ont été attribuées selon une distribution cyclique). Soit $n_{first}^{(k)}(j)$ la première tranche que ce $j^{ème}$ processeur doit résoudre à la $k^{ème}$ itération. Ils résoudra alors les tranches :

$$n \in \{n_{first}^{(k)}(j)\,,\, n_{first}^{(k)}(j) + n_p\,,\, n_{first}^{(k)}(j) + 2n_p\,,\, \cdots\,,\, n_{first}^{(k)}(j) + l_{div}^{(k)} n_p\},$$

en utilisant les prédictions $Y_{n-1}^{p^{(k)}}$, et en évaluant le saut relatif entre les valeurs corrigées Y_n^c de fin de tranche et leurs prédictions $Y_n^{p^{(k)}}$ (qu'il connaît, puisque les prédictions sont dupliquées sur tous les processeurs) :

$$GAP_n = \frac{Y_n^c - Y_n^{p^{(k)}}}{\max\left(\|Y_n^c\|_\infty\,,\, \left\|Y_n^{p^{(k)}}\right\|_\infty\right)}.$$

Le $j^{ème}$ processeur continuera à résoudre ses tranches tant que $n \leq N$ et que le test de convergence est vérifié :

$$\|GAP_n\|_\infty \leq \epsilon_{tol}^g,$$

où ϵ_{tol}^g est une tolérance donnée.
La première qui diverge sera la dernière tranche que le processeur va résoudre, c'est-à-dire la tranche $n_{div}^{(k)}(j) = n_{first}^{(k)}(j) + l_{div}^{(k)} n_p$.
Quand tous les processeurs auraient arrêté leurs calculs, on peut déduire le numéro de la

première tranche à avoir divergé (parmi tous les processeurs) à la $k^{ème}$ itération :

$$n_s^{(k)} = \left(\min_{1 \leq j \leq n_p} n_{div}^{(k)}(j) \right).$$

Il s'ensuit que les tranches :

$$n \in \{n_s^{(k-1)} + 1, n_s^{(k-1)} + 2, \cdots, n_s^{(k)}\},$$

sont considérées avoir convergé, c'est-à-dire que les valeurs calculées $\{Y_n^c\}$ et $\{T_n^c\}$ sont considérées bonnes. Ainsi, à la fin de la $k^{ème}$ itération, le problème est complètement résolu jusqu'à la tranche $n_s^{(k)}$.

Il faut remarquer qu'il y a au moins une tranche qui converge à chaque itération, la tranche $n_s^{(k-1)} + 1$, qui commence avec une valeur exacte (non prédite) de la solution. Comme pour tous les schémas du genre, on a alors :

$$n_s^{(k)} \geq n_s^{(k-1)} + 1.$$

Théoriquement, chaque processeur devrait communiquer, après avoir résolu chaque $n^{ème}$ tranche et réussi le test de convergence, avec le processeur qui a résolu la tranche précédente et ce, afin de valider la valeur prédite Y_{n-1}^p pour en déduire la validation de Y_n^c and Y_n^p et continuer à résoudre. Mais en pratique, la distribution cyclique des tranches fait que les prédictions cessent d'être bonnes presque en même temps pour tous les processeurs, créant une sorte de synchronisation des calculs parallèles. Le seul risque est alors que chaque processeur résoud une tranche de plus, à chaque itération, en économisant cependant beaucoup de communications.
Il faut enfin remarquer que l'algorithme RaPTI a la caractéristique de faire arrêter les calculs sur chaque processeur, dès qu'une de ses tranches diverge.
L'étape de correction décrite ci-dessus est implémentée conformément à la procédure SOLVE_MY_SLICES_IN_PARALLEL, donnée en annexe 1.

Etape de communication :
Rappelons que la duplication des calculs séquentiels (d'initialisation et de prédiction) réduit le nombre de communications à une seule étape de communication en fin de chaque itération. Chacune de ces étapes se fait comme suit.

1. A la fin de la phase de correction de la $k^{ème}$ itération, chaque $j^{ème}$ processeur j $(2 \leq j \leq n_p)$ envoie au processeur 1 (maître) le numéro $n_{div}^{(k)}(j)$ de sa première tranche à avoir divergé, ainsi que les tailles $\{\Delta T_n^c = \beta_n^p s_n^c\}$ des tranches qu'il a calculées au cours de cette itération.

2. Le processeur 1 reçoit les messages et évalue ce qui suit :

   - Le numéro $n_s^{(k)}$ de la dernière tranche à avoir globalement convergé (sur tous les processeurs) :
   $$n_s^{(k)} = \left(\min_{1 \leq j \leq n_p} n_{div}^{(k)}(p) \right).$$
   - Les temps $\{T_n^c\}$ de début de tranches, pour $n_s^{(k-1)} + 1 \leq n \leq n_s^{(k)}$, en utilisant la récurrence:
   $$T_n^c = T_{n-1}^c + \Delta T_n^c,$$

et en commençant par $T^c_{n_s^{(k-1)}}$, qui est connu depuis l'itération précédente.

Ces calculs sont faits en implémentant sur le processeur 1 la procédure DETERMINE_$n_s^{(k)}$_AND_$\{T_n^c\}$, donnée en annexe 1.
Le processeur 1 envoie ensuite ses résultats à tous les autres processeurs.

3. Tous les processeurs j ($j \neq 1$) reçoivent $n_s^{(k)}$ et $\{T_n^c\}_{n\in\{n_s^{(k-1)}+1,\cdots,n_s^{(k)}\}}$, et font ce qui suit :

   - Chacun déduit l'étendue $n_{conv}^{(k)}$ de convergence de la $k^{ème}$ itération, i.e. le nombre de tranches ayant globalement convergé au cours de cette itération :

   $$n_{conv}^{(k)} = n_s^{(k)} - n_s^{(k-1)}.$$

   - Chaque processeur calcule le vecteur temps réel de la solution sur chacune des tranches $n \in \{n_s^{(k-1)}, \cdots, n_s^{(k)}\}$ qu'il avait calculées :

   $$\forall t \in [T^c_{n-1}, T^c_n], \quad t = T^c_{n-1} + \beta^p_n s^c_n.$$

   - – Si $T_{n_s^{(k)}} < T$, alors tous les processeurs réestiment N (obtenant $N^{(k)}$), en se basant sur $T_{n_s^{(k)}}$. De plus, chaque processeur ayant une des dernières n_{prec} slices envoie à tous les autres processeurs ses valeurs Y_n^c de fin de tranche de la solution, puisqu'ils en ont besoin pour les nouvelles prédictions.
     – Si $T_{n_s^{(k)}} \geq T$, alors $N^{(k)} = n_s^{(k)}$ et le schéma itératif prend fin.

La figure (4.7) illustre l'étape de communication décrite ci-dessus.

Intégration numérique :
Vu l'approche locale caractérisant le calcul en tranches du problème, avec redimensionnement et certaines propriétés de similarité qui en découlent, une méthode numérique explicite a été utilisée, nommément : la méthode explicite de Runge Kutta d'ordre 4.
Par ailleurs, un pas de temps redimensionné τ est calculé, en début de chaque tranche, de sorte à assurer une tolérance donnée sur la précision des calculs. La résolution numérique de chacun des systèmes redimensionnés est alors faite sur une grille de temps fine de pas constant τ, ce pas étant toutefois raffiné en fin de tranche en vue d'atteindre la condition EOS avec une précision donnée. Il ne faut pas oublier que, même lorsque les pas de temps redimensionnés sont constants, ils sont modulés par les paramètres β_n et se traduisent en grille de temps réel non régulière.

Enfin, on peut trouver en annexe 2 :
- une analyse rapide du coût de communication de l'algorithme RaPTI,
- la méthode utilisée pour l'évaluation de l'efficacité et de l'accélération des calculs ("speed-up"),
- une description de l'architecture parallèle et du langage de programmation utiliés lors de l'implémentation de RaPTI.

III- Applications Numériques

a) Problème de membrane

On considère le système d'équations différentielles ordinaires de second ordre :

$$(Memb.) \qquad \begin{cases} y'' - b|y'|^{q-1}\, y' \;+\; |y|^{p-1}\, y = 0, \;\; t > 0, \\ y(0) = y_{1,0}, \\ y'(0) = y_{2,0}, \end{cases}$$

où $y : [0, T] \longrightarrow \mathbb{R}$ $(T \leq \infty)$.
Ce système modélise le mouvement d'une membrane reliée à un ressort. Le terme non-linéaire en y décrit la rigidité du ressort et celui en y' décrit, selon le signe de b, une excitation (si $b > 0$) ou un frottement (si $b < 0$). Ce dernier cas correspond à un système dissipatif admettant une solution globale sur $[0, \infty)$ ([55], [56]).
Lorsque $b > 0$, l'excitation du mouvement entraîne une explosion de la solution, i.e. l'existence de $T_b \leq \infty$ tel que : $\lim_{t \to T_b} |y(t)| = \lim_{t \to T_b} |y'(t)| = \infty$.
Ce cas a été largement étudié par Souplet et al dans [57], [58], [59] et [60], où l'existence et l'unicité de la solution ont été discuté et le comportement de la solution déterminé dans certains cas. Ils ont par exemple prouvé que dans le cas où $b = 1$ et $p, q > 1$, il existe deux valeurs critiques $q = p$ et $q = \frac{2p}{p+1}$ qui déterminent dans le plan (p, q) trois régions de comportement différent (voir figure 5.1).
Dans [49] et [51], nous avons appliqué la méthode de calculs en tranches avec redimensionnement au cas où $p > 1$ et $q = \frac{2p}{p+1}$, dont la solution explose en temps fini, de façon oscillatoire ou non oscillatoire (selon la valeur de b). Nous avons ainsi réussi à contrôler la raideur du problème et obtenu des valeurs très précises de la solution et du temps fini d'explosion.
Dans cette thèse, nous nous intéressons au cas :

$$0 < p \leq q \leq \frac{2p}{p+1} \leq 1,$$

pour lequel nos essais numériques ont montré l'existence d'une solution qui explose en temps infini, y et y' présentant un comportement oscillatoire, i.e.:

1. $\lim_{t \to \infty} |y(t)| = \lim_{t \to \infty} |y'(t)| = \infty$, et
2. $y(t)$ et $y'(t)$ ayant un nombre infini de racines dans l'intervalle $[0, \infty)$.

Dans le plan de phase (y, y'), la trajectoire de la solution tourne en spirale en fuyant vers l'infini (voir figure 5.1).

En posant $Y_1 = y$ et $Y_2 = y'$, le problème se ramène à un système d'EDOs de premier ordre :

$$(S) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t > 0 \\ Y(0) = Y_0, \end{cases}$$

où $Y : [0, \infty) \to \mathbb{R}^2$, $Y_0 = \begin{pmatrix} Y_{1,0} \\ Y_{2,0} \end{pmatrix}$ et $F \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} Y_2 \\ b|Y_2|^{q-1}\, Y_2 \;-\; |Y_1|^{p-1}\, Y_1 \end{pmatrix}$.

Choix d'une condition EOS et des paramètres $\{\beta_n\}$:
La solution du problème $(Memb.)$ étant de nature oscillatoire, il faudrait terminer une tranche de temps à chaque fois que la trajectoire de la solution finit une rotation complète

dans le plan de phase (Y_1, Y_2).
Une façon très particulière est de le faire à chaque fois que la trajectoire de la solution coupe la courbe d'équation $Y_2 = |Y_1|^{\frac{p+1}{2}}$ dans le premier quadrant du plan de phase (Y_1, Y_2) (voir figure 5.3).
Cette condition EOS est alors régie par la fonction L définie par :

$$\begin{cases} L\left[Y(t)\right] = 0 & \text{si } Y_2(t) = |Y_1(t)|^{\frac{p+1}{2}} \quad \text{et} \quad Y_1(t) > 0, \\ L\left[Y(t)\right] = 1 & \text{si } Y_2(t) \neq |Y_1(t)|^{\frac{p+1}{2}} \quad \text{ou} \quad Y_1(t) \leq 0. \end{cases}$$

Le problème devient ainsi équivalent à la suite de problèmes de tir suivants :

$$(S_n) \qquad \begin{cases} \frac{dY_1}{dt} = Y_2, \\ \frac{dY_2}{dt} = b|Y_2|^{q-1}\, Y_2 \; - \; |Y_1|^{p-1}\, Y_1, \\ Y_1(T_{n-1}) = Y_{n-1,1}, \\ Y_2(T_{n-1}) = Y_{n-1,2}, \\ L\left[Y(T_n)\right] = 0 \quad \text{et} \quad \forall t \in (T_{n-1}, T_n), \;\; L\left[Y(t)\right] \neq 0. \end{cases}$$

Notons que la condition de fin de tranche choisie réalise la condition ($Nonzeroness$) sur les $\{Y_n\}$. On procède alors au changement de variable ($ChVar0$), avec le choix suivant des β_n :

$$\beta_n = |Y_{n-1,1}|^{\frac{1-p}{2}} = |Y_{n-1,2}|^{\frac{1-p}{p+1}}\,.$$

On obtient la suite de problèmes de tir redimensionnés suivants :

$$(S'_n) \qquad \begin{cases} \frac{dZ_{n,1}}{ds} = G_{n,1}(Z_n), & 0 < s \leq s_n, \\ \frac{dZ_{n,2}}{ds} = G_{n,2}(Z_n), & 0 < s \leq s_n, \\ Z_{n,1}(0) = Z_{n,2}(0) = 0, \\ L_z\left[Z_n(s_n)\right] = 0 \quad \text{et} \quad \forall s \in (0, s_n), \;\; L_z\left[Z_n(s)\right] \neq 0, \end{cases}$$

où :

$$\begin{cases} G_{n,1}(Z_n) = 1 + Z_{n,2}, \\ G_{n,2}(Z_n) = b\gamma_n|1 + Z_{n,2}|^{q-1}(1 + Z_{n,2}) - \; |1 + Z_{n,1}|^{p-1}(1 + Z_{n,1}), \end{cases}$$

avec

$$\gamma_n = |Y_{n-1,1}|^{\frac{(p+1)(q-1)}{2} + \frac{1-p}{2}}$$

et

$$\begin{cases} L_z\left[Z_n(s)\right] = 0 & \text{si } 1 + Z_{n,2}(s) = |1 + Z_{n,1}(s)|^{\frac{p+1}{2}} \quad \text{et} \quad 1 + Z_{n,1}(s) > 0 \\ L_z\left[Z_n(s)\right] = 1 & \text{si } 1 + Z_{n,2}(s) \neq |1 + Z_{n,1}(s)|^{\frac{p+1}{2}} \quad \text{ou} \quad 1 + Z_{n,1}(s) \leq 0 \end{cases}$$

Propriétés d'invariance et de similarité asymptotique :
On montre facilement que, pour les choix adoptés de condition EOS et paramètres $\{\beta_n\}$, on obtient les propriétés de similarité suivantes :

1. Si $\forall p \leq 1$, $q = \frac{2p}{p+1}$, alors les systèmes (S'_n) sont invariants et equivalents, pour tout n, au problème de tir unique :

$$(S'_1) \qquad \begin{cases} \frac{dZ_{1,1}}{ds} = (1 + Z_{1,2}), & 0 < s \leq s_1, \\ \frac{dZ_{1,2}}{ds} = b|1 + Z_{1,2}|^{q-1}(1 + Z_{1,2}) \\ \qquad\qquad\qquad - |1 + Z_{1,1}|^{p-1}(1 + Z_{1,1}), \\ Z_{1,1}(0) = Z_{1,2}(0) = 0, \\ L_z\left[Z_n(s_n)\right] = 0 \quad \text{et} \quad \forall s \in (0, s_n), \;\; L_z\left[Z_n(s)\right] \neq 0. \end{cases}$$

2. Si $0 < p \leq q < \frac{2p}{p+1} \leq 1$, alors les systèmes (S'_n) sont asymptotiquement similaires à un problème de tir limite défini par :

$$(S'_L) \qquad \begin{cases} \frac{dZ_{L,1}}{ds} = 1 + Z_{L,2}, & 0 < s \leq s_L, \\ \frac{dZ_{L,2}}{ds} = -|1 + Z_{L,1}|^{p-1}(1 + Z_{L,1}), \\ Z_{L,1}(0) = Z_{L,2}(0) = 0, \\ L_z\left[Z_n(s_n)\right] = 0 \quad \text{et} \quad \forall s \in (0, s_n), \;\; L_z\left[Z_n(s)\right] \neq 0. \end{cases}$$

Les figures 5.5 et 5.6 illustrent un cas d'invariance et un cas de similarité asymptotique, respectivement, dans le plan de phase (Z_1, Z_2).

Résultats numériques en cas d'invariance :
Rappelons que, en cas d'invariance, tout parallélisme devient inutile (malgré une propriété des ratios parfaite). L'algorithme RaTI utilise la propriété d'invariance et sera appliqué, sur un seul processeur. Cependant, on a d'abord appliqué l'algorithme parallèle RaPTI, à seul titre de référence: comme prévu, il a abouti à d'excellents "speed-ups" (voir tableau 5.2 et figure 5.7). Puis on a appliqué l'algorithme RaTI, qui utilise séquentiellement la propriété d'invariance, *sur un seul processeur*. Le résultat est l'obtention de speed-ups de l'ordre de 50, calculés en comparant au temps de calcul nécessaire pour résoudre le même nombre de tranches sans utiliser la propriété d'invariance. (voir tableau 5.3)

Résultats numériques en cas de similarité asymptotique :
Comme l'indique le tableau 5.4, ce problème donne une propriété des ratios notable.
Un modèle polynomial de degré 2, utilisant les 3 derniers ratios, a été adopté et a donné d'excellentes prédictions. Le résultat a été d'obtenir un nombre d'itérations très petit, comparé au nombre total de tranches à résoudre.
L'algorithme RaPTI a été testé sur 8 cas de combinaisons des paramètres p et q du problème et pour des tolérances différentes sur la propriété des ratios et les sauts ($\epsilon_{tol}^{n_s}$ et ϵ_{tol}^{g} respectivement). Dans tous les cas, on a obtenu d'excellents speed-ups (voir les tableaux 5.5 et 5.6 et les figures 5.8 et 5.9).
Par ailleurs, nous avons essayé d'étudier l'impact du choix des tolérances $\epsilon_{tol}^{n_s}$ et ϵ_{tol}^{g} sur les speed-ups, les efficacités et la précision des calculs (voir figures 5.10, 5.11 et 5.12).
Après cette étude comparative, on a pu remarquer que :

- La précision des calculs n'est presque pas affectée par les variations de $\epsilon_{tol}^{n_s}$, mais est bien améliorée en prenant des plus petites tolérances ϵ_{tol}^{g}.
- Le speed-up n'est presque pas affecté par les variations de ϵ_{tol}^{g}, mais est bien améliorée en prenant de plus grandes tolérances $\epsilon_{tol}^{n_s}$.

Ainsi, et comme notre but est d'avoir à la fois de bons speed-ups et une bonne précision des calculs, on en déduit que le choix optimal des tolérances consiste à minimiser ϵ_{tol}^{g} et à maximiser $\epsilon_{tol}^{n_s}$, dans les limites du possible bien sûr.

b) Problème de réaction-diffusion

On considère le problème de réaction-diffusion défini par :

$$(RD) \qquad \begin{cases} \frac{\partial u}{\partial t} - \Delta u^m = a u^p, \; x \in \Omega \subset \mathbb{R}^d, \; t > 0, \\ u(x,t) = 0, \; x \in \partial\Omega, \; t \geq 0, \\ u(x,0) = u_0(x) > 0, \; x \in \Omega. \end{cases}$$

où $a > 0$, $m > 0$, $p > 0$, et Δ est l'opérateur Laplacien.
u est une concentration qui vérifie la propriété de positivité, Δu^m est le terme de diffusion et au^p est le terme de réaction. Il est à noter que le comportement de la solution varie avec les valeurs de m et p ainsi qu'avec les conditions initiales (voir [61]).
Dans cette thèse, on s'intéresse au cas où :

$$0 < m \leq p \leq 1,$$

dont la solution peut, pour certaines conditions initiales, croître vers l'infini de façon monotone, comme prouvé dans [61]. De plus, nos essais numériques ont montré que chacune des composantes de la solution a une croissance monotone et que l'explosion a lieu sur tout le domaine.
On commence d'abord par procéder au changement de variables $v = u^m$. En posant $q = \frac{1}{m}$, le problème s'écrit :

$$\begin{cases} \frac{\partial v}{\partial t} = \frac{1}{q}\frac{1}{v^{q-1}}\Delta v + \frac{a}{q}v^{pq-q+1}, \; x \in \Omega \subset \mathbb{R}^d, \; t > 0, \\ v(x,t) = 0, \; x \in \partial\Omega, \; t \geq 0, \\ v(x,0) = v_0(x) > 0, \; x \in \Omega. \end{cases}$$

où $v_0(x) = [u_0(x)]^m = [u_0(x)]^{\frac{1}{q}}$, et $0 < \frac{1}{q} \leq p \leq 1$, entraînant $q \geq 1$ et $2-q \leq pq-q+1 \leq 1$.

Ensuite, on applique une semi-discrétisation dans l'espace et sur l'intérieur du domaine Ω, de dimension K, conduisant à un système aux dérivées ordinaires. On utilise pour cela la discrétisation de type différences finies qui approxime $-\Delta$ par une matrice creuse $A \in \mathbb{R}^{K\times K}$ symétrique définie positive. La solution $v(x,t)$ et la valeur initiale $v_0(x)$ sont approximées par les vecteurs :

$$v(x,t) \approx Y(t) = \begin{pmatrix} Y_1(t) \\ Y_2(t) \\ \vdots \\ Y_K(t) \end{pmatrix} \quad \text{et} \quad v_0(x) \approx Y_0 = \begin{pmatrix} Y_{0,1} \\ Y_{0,2} \\ \vdots \\ Y_{0,K} \end{pmatrix}.$$

Dans cette partie, on utilisera les notations $V.^\alpha$ et D_V définies comme suit :

$$\forall \alpha \in \mathbb{R}, \quad \forall V = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{pmatrix} \in \mathbb{R}^K \text{ avec } \forall i, \; V_i > 0, \qquad V.^\alpha = \begin{pmatrix} V_1^\alpha \\ V_2^\alpha \\ \vdots \\ V_K^\alpha \end{pmatrix}$$

et

$$\forall V = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{pmatrix} \in \mathbb{R}^K, \qquad D_V = \begin{pmatrix} V_1 & 0 & \cdots & 0 \\ 0 & V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_K \end{pmatrix} \in \mathbb{R}^{K\times K}.$$

Après semi-discrétisation, le problème (RD) devient équivalent au problème de Cauchy autonome:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = F(Y), \quad t > 0, \\ Y(0) = Y_0, \end{cases}$$

où $Y : [0, \infty) \to \mathbb{R}^K$ et :

$$F(Y) = F_{diff}(Y) + F_{reac}(Y),$$

$F_{diff}(Y)$ and $F_{reac}(Y)$ correspondant respectivement aux termes de diffusion et réaction du problème :

$$F_{diff}(Y) = -\frac{1}{q} D^{-1}_{Y.^{q-1}} AY,$$

$$F_{reac}(Y) = \frac{a}{q} Y.^{pq-q+1}.$$

Choix d'une condition EOS et des paramètres $\{\beta_n\}$:
La solution u du problème (RD) étant positive, il s'ensuit que la solution Y du problème (S) ci-dessus est positive dans toutes ses composantes : $\forall t > 0,\ \forall i,\ \ Y_i(t) > 0$.
Cette solution étant de plus de nature explosive, on pourra terminer une tranche de temps de sorte que la valeur absolue de la variation relative de chacune des composantes de $Y(t)$ ne dépasse pas un seuil $S > 0$ choisi (les $\{Y_n\}$ vérifiant la condition $(Nonzeroness)$). Cette condition EOS est alors régie par la famille de fonctions $\{E_n\}$ données, pour tout $n > 0$, par :

$$E_n\left[Y(t)\right] = \left\| D_n^{-1}\left(Y(t) - Y_{n-1}\right)\right\|_\infty - S.$$

Le problème (RD) devient ainsi équivalent à la suite de problèmes de tir suivants :

$$(S_n) \qquad \begin{cases} \frac{dY}{dt} = F_{diff}(Y) + F_{reac}(Y), \\ Y(T_{n-1}) = Y_{n-1}, \\ E_n\left[Y(T_n)\right] = 0 \text{ et } \forall t \in (T_{n-1}, T_n),\ \ E_n\left[Y(t)\right] \neq 0. \end{cases}$$

On procède alors au changement de variable $(ChVar0)$, avec le choix suivant des β_n :

$$\beta_n = \frac{1}{\left\|\left(Y_{n-1}\right).^{pq-q}\right\|_\infty}.$$

On obtient la suite de problèmes de tir redimensionnés suivants :

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n) \quad 0 < s \leq s_n, \\ Z_n(0) = 0, \\ \forall s < s_n,\ \left\|Z_n(s)\right\|_\infty < S,\ \text{et}\ \ \left\|Z_n(s_n)\right\|_\infty = S, \end{cases}$$

où :

$$G_n(Z_n) = G_{n_{diff}}(Z_n) + G_{n_{reac}}(Z_n),$$

avec :

$$G_{n_{diff}}(Z_n(s)) = \ -\ \frac{1}{q} \frac{1}{\left\|\left(Y_{n-1}\right).^{pq-q}\right\|_\infty} D^{-1}_{(Y_{n-1}).^q} D^{-1}_{[\mathbf{1}+Z_n(s)].^{q-1}} A D_{Y_{n-1}}\left[\mathbf{1} + Z_n(s)\right]$$

$$G_{n_{reac}}(Z_n(s)) = \frac{a}{q} \frac{1}{\left\|\left(Y_{n-1}\right).^{pq-q}\right\|_\infty} D_{(Y_{n-1}).^{pq-q}}\left[\mathbf{1} + Z_n(s)\right].^{pq-q+1}$$

Propriété de similarité asymptotique :
On montre que, pour les choix adoptés de condition EOS et paramètres $\{\beta_n\}$, et sous certaines conditions, les problèmes redimensionnés (S'_n) sont asymptotiquement similaires à un problème limite donné par :

1. dans le cas non linéaire, $0 < m < p \leq 1$:

$$(S_L) \qquad \begin{cases} \frac{dZ_L}{ds} = G_L(Z_L), \quad 0 < s \leq s_L, \\ Z_L(0) = 0, \\ \forall s < s_L,\ \|Z_L(s)\|_\infty < S,\ \text{et}\ \|Z_L(s_L)\|_\infty = S, \end{cases}$$

où :

$$G_L(Z_L) = \frac{a}{q}\left[\mathbf{1} + Z_L(s)\right].^{pq-q+1}.$$

2. dans le cas linéaire, $m = p = 1$:

$$(S_L) \qquad \begin{cases} \frac{dZ_L}{ds} = G_L(Z_L),\ 0 < s \leq s_L, \\ Z_L(0) = 0, \\ \forall s < s_n,\ \|Z_n(s)\|_\infty < S,\ \text{et}\ \|Z_n(s_n)\|_\infty = S, \end{cases}$$

où :

$$G_L(Z_L) = \lambda \exp(s\lambda)\,\mathbf{1},$$

Les figures 6.4 et 6.7 illustrent cette similarité asymptotique dans le plan $(s, Z_{n,i})$, et dans deux cas non linéaire et linéaire respectivement.

Résultats numériques :
Comme l'indiquent les tableaux 6.2, 6.3, 6.4, 6.8 et 6.9, ce problème aboutit à une propriété des ratios notable, dans les deux cas linéaire et non linéaire.
Un modèle polynomial de degré 2, utilisant les 3 derniers ratios, a été adopté et a donné d'excellentes prédictions. Le résultat a été une convergence très rapide, souvent avec une itération unique.
L'algorithme RaPTI a été testé sur 15 cas de combinaisons des paramètres p et m du problème, dans le cas non linéaire, et sur 4 valeurs du coefficient a combinées avec deux conditions initiales différentes, dans le cas linéaire. Dans les deux cas, on a obtenu de bons speed-ups (voir les tableaux 6.6, 6.7, 6.10 et les figures 6.3.2, 6.6 et 6.4.2).

c) Trajectoire de satellite

Les trajectoires de satellite sont déterminées en résolvant le problème de second ordre découlant des lois de Newton :

$$\vec{F} = m\ddot{\vec{r}}$$

où m est la masse de la terre, $\vec{r}$ est le vecteur-position du satellite (à partir du centre de la terre), $\ddot{\vec{r}}$ est le vecteur accélération et $\vec{F}$ est la résultante des forces appliquées sur le satellite et qui sont de natures diverses (gravitationnelle ou de surface).
Dans un mouvement keplerien, $\vec{F}$ se réduit à la seule composante centrée de la gravitation terrestre : $\vec{F} = -GM\frac{m}{\|\vec{r}\|^3}\vec{r} = -\mu\frac{m}{\|\vec{r}\|^3}\vec{r}$, où G est la constante universelle de gravitation, M la masse de la terre et $\mu = GM = 3986005 \times 10^8 m^3/s^2$. Dans ce cas, les lois de Kepler s'appliquent, le mouvement du satellite est périodique et sa trajectoire décrit une ellipse située dans le plan keplerien (qui est entièrement déterminé par les conditions initiales).
Cependant, beaucoup de forces perturbent ce mouvement. La plus importante de ces forces est la "force en J_2" qui est due à l'aplatissement de la terre.
Erhel et Rault ont proposé dans [7] et [62] un schéma parallèle intéressant, qui est une variante des méthode de tirs multiples, en tenant compte de toutes les forces perturbatrices. Dans cette thèse, nous appliquons l'algorithme RaPTI sur le modèle simplifié en J_2 qui ne

tient compte que du potentiel de gravitation terrestre suivant :

$$u(J_2) = \frac{\mu}{r} + \frac{\mu}{r}\left(\frac{r_{eq}}{r}\right)^2 J_2 P_2(\sin\varphi) = U_K + U_P$$

où $r_{eq} = 6378.137km$ est le rayon équatorial, $J_2 = -11.10^{-4}$ est l'harmonique zonale d'ordre 2, $r = \|\vec{r}\|_2$ est le module du vecteur-position, φ est la latitude et $P_2(\sin\varphi) = \frac{3}{2}\sin^2\varphi - \frac{1}{2}$ est le polynôme du second degré de Legendre (en $\sin\varphi$) (en cas de mouvement keplerien, ce potentiel se réduit à $u(J_2) = U_K = \frac{\mu}{r}$).
Le problème en J_2 est alors décrit par le le système :

$$(Sat.) \qquad \begin{cases} \ddot{\vec{r}}(t) = \vec{f}(\vec{r}), \quad t > 0, \\ \vec{r}(0) = \vec{r_0}, \\ \dot{\vec{r}}(0) = \vec{\dot{r}}_0, \end{cases}$$

avec $\vec{f}(\vec{r}) = \vec{\nabla} u(J_2) = \vec{\nabla}(U_K + U_P) = \vec{f_K}(\vec{r}) + \vec{f_P}(\vec{r})$, où $\vec{f_K}(\vec{r}) = \vec{\nabla} U_K$ et $\vec{f_P}(\vec{r}) = \vec{\nabla} U_P$ sont les composantes kepleriennes et perturbatrices de $\vec{\nabla} u(J_2)$ respectivement.
L'expression explicite de $\vec{f_K}(\vec{r})$ est :

$$\vec{f_K}(\vec{r}) = \vec{f_K}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{\partial U_K}{\partial x} \\ \frac{\partial U_K}{\partial y} \\ \frac{\partial U_K}{\partial z} \end{pmatrix} = \begin{pmatrix} -\mu\frac{x}{r^3} \\ -\mu\frac{y}{r^3} \\ -\mu\frac{z}{r^3} \end{pmatrix}.$$

et celle de $\vec{f_P}(\vec{r})$ est donnée dans le repère géocentrique inertiel ECI par :

$$\left[\vec{f_P}\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{ECI}\right]_{ECI} = \begin{pmatrix} \frac{\partial U_P}{\partial x} \\ \frac{\partial U_P}{\partial y} \\ \frac{\partial U_P}{\partial z} \end{pmatrix}_{ECI} = \begin{pmatrix} \frac{3\mu J_2 r_{eq}^2}{2}\left(1 - 5\frac{z^2}{r^2}\right)\frac{x}{r^5} \\ \frac{3\mu J_2 r_{eq}^2}{2}\left(1 - 5\frac{z^2}{r^2}\right)\frac{y}{r^5} \\ \frac{3\mu J_2 r_{eq}^2}{2}\left(3 - 5\frac{z^2}{r^2}\right)\frac{z}{r^5} \end{pmatrix}_{ECI}$$

où $r = \sqrt{x^2 + y^2 + z^2}$.
Dans ce mouvement perturbé, la trajectoire du satellite est, à tout instant, tangente à une ellipse intantanée (ellipse osculatrice) définie par les valeurs instantanées des éléments orbitaux variables. Nous définissons alors un système $IPQW$ de coordonnées cartésiennes dont l'origine est le centre O de la terre et le plan (xOy) est le plan keplerien P_0 correspondant aux conditions initiales, la direction (Oz) exprimant alors l'effet de la perturbation:
- l'axe (Ox) est dirigé par le vecteur reliant le centre de la terre au périgée de l'ellipse initiale,
- l'axe (Oy) est l'image, par une rotation de 90° et dans le plan P_0, de l'axe (Ox),
- l'axe (Oz) complète le trièdre direct (Ox, Oy, Oz).

Dans ce système de coordonnées, l'expression explicite de $\vec{f_P}(\vec{r})$ est donnée par:

$$\left[\vec{f_P}\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{IPQW}\right]_{IPQW} = A\left[\vec{f_P}\left(A^T\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{IPQW}\right)\right]_{ECI},$$

où A est la matrice orthogonale de transformation qui permet de convertir les coordonnées cartésiennes en repère ECI à celles en repère IPQW.

Après rabaissement d'ordre, la résolution du problème $(Sat.)$ se ramène à celle d'un problème de premier ordre de la forme :

$$(S) \qquad \left\{ \begin{array}{l} \frac{dY}{dt} = F\left(Y\right), \quad t > 0, \\ Y(0) = Y_0, \end{array} \right.$$

où $Y : [0, \infty) \to \mathbb{R}^6$, avec : $\quad Y = \begin{pmatrix} \vec{r} \\ \dot{\vec{r}} \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{pmatrix}, \quad Y_0 = \begin{pmatrix} \vec{r_0} \\ \dot{\vec{r}_0} \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ \dot{x_0} \\ \dot{y_0} \\ \dot{z_0} \end{pmatrix}$

et $F(Y) = F_K(Y) + F_P(Y)$, où $F_K(Y) = \begin{pmatrix} \dot{\vec{r}} \\ \vec{f_K}\left(\vec{r}\right) \end{pmatrix}$ et $F_P(Y) = \begin{pmatrix} \vec{0} \\ \vec{f_P}\left(\vec{r}\right) \end{pmatrix}$.

Application à un mouvement keplerien: Cas d'invariance

Choix d'une condition EOS et redimensionnement :

Dans ce mouvement périodique, la trajectoire du satellite décrit une ellipse, dans le plan xOy, dont les paramètres sont entièrement déterminés par les conditions initiales et le problème (S) ci-dessus se réduit à :

$$(S) \qquad \left\{ \begin{array}{l} \frac{dY}{dt} = F_K(Y), \quad t > 0 \\ Y(0) = Y_0, \end{array} \right. \qquad \text{avec} \qquad F_K(Y) = \begin{pmatrix} Y_4 \\ Y_5 \\ 0 \\ -\mu \frac{Y_1}{R^3} \\ -\mu \frac{Y_2}{R^3} \\ 0 \end{pmatrix} \quad \text{et} \quad R = \sqrt{Y_1^2 + Y_2^2}.$$

Le choix évident de condition EOS est de terminer une tranche de temps à chaque fois que l'angle polaire, dans le plan xOy, retrouve sa valeur initiale, le satellite ayant alors décrit toute l'ellipse et retrouvé sa position initiale. Le mouvement étant périodique, les paramètres $\{\beta_n\}$ de redimensionnement du temps seront constants et égaux à 1.

En procédant au changement de variables $(ChVar)$, le problème devient alors équivalent à la suite de problèmes de tir redimensionnés suivants :

$$(S'_n) \qquad \left\{ \begin{array}{l} \frac{dZ_n}{ds} = G_{n,K}(Z_n), \ s > 0, \\ Z_n(0) = 0, \\ \tilde{\theta}\left[Z_n(s_n)\right] = 2\pi, \quad \text{et} \quad \forall s \in (0, s_n), \ \tilde{\theta}\left[Z_n(s)\right] < 2\pi. \end{array} \right.$$

où $G_{n,K}(Z_n) = \frac{dZ_n}{ds} = D_n^{-1} \begin{pmatrix} Y_4 \\ Y_5 \\ 0 \\ -\mu \frac{Y_1}{R^3} \\ -\mu \frac{Y_2}{R^3} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{Y_{n-1,4}}{Y_{n-1,1}} \left[1 + Z_{n,4}(s)\right] \\ \frac{Y_{n-1,5}}{Y_{n-1,2}} \left[1 + Z_{n,5}(s)\right] \\ 0 \\ -\mu \left(\frac{Y_{n-1,1}}{Y_{n-1,4}}\right) \frac{1+Z_{n,1}(s)}{R^3} \\ -\mu \left(\frac{Y_{n-1,2}}{Y_{n-1,5}}\right) \frac{1+Z_{n,2}(s)}{R^3} \\ 0 \end{pmatrix}.$

avec $R = \sqrt{Y_1^2 + Y_2^2} = \sqrt{Y_{n-1,1}^2 \left[1 + Z_{n,1}(s)\right]^2 + Y_{n-1,2}^2 \left[1 + Z_{n,2}(s)\right]^2}$.

Propriété d'invariance :
Le mouvement étant périodique, on a : $\forall n,\ \forall i,\ Y_{n,i} = Y_{0,i}$. Ceci implique, pour tout n : $G_n(.) = G_1(.)$, d'où l'invariance des problème (S'_n).

Résultats numériques en ce cas d'invariance :
Dans les tableaux (7.1) et (7.2), on trouve les résultats d'application de l'algorithme séquentiel RaTI (propre aux cas d'invariance), dans 12 conditions initiales différentes. Chacune des conditions initiales est donnée par ses six éléments orbitaux kepleriens $(a_0, e_0, i_0, \omega_0, \Omega_0, M_0)$. L'implémentation a été faite *sur un seul processeur* et a mené à d'énormes speed-ups, de l'ordre de 55, calculés en comparant au temps de calcul nécessaire pour résoudre le même nombre de tranches sans utiliser la propriété d'invariance.

Application à un mouvement perturbé en J_2: Cas de similarité faible
Choix d'une condition EOS et redimensionnement :
Dans ce cas de mouvement non keplerien, et donc non planaire, la trajectoire est définie par une ellipse osculatrice instantanée dont les paramètres sont continument variables (voir figure 7.7). Puisque toutes ces ellipses instantanées ont le centre de la terre pour un de leurs foyers, l'orbite du satellite traversera régulièrement le plan xOy (qui est le plan keplerien correspondant aux conditions initiales et que le satellite quitte à son départ). Une condition EOS possible est alors de terminer chaque $n^{ème}$ tranche de temps $[T_{n-1}, T_n]$ lorsque le satellite traverse le plan xOy, après une rotation complète, c'est-à-dire lorsque $Y_3(t) = 0$ pour la deuxième fois après l'avoir été à $t = T_{n-1}$. Cette condition EOS est ainsi régie par la fonction L définie par :

$$\begin{cases} L\left[Y(t)\right] = 0 & \text{si } Y_3(t) = 0 \quad \text{pour la } 2^{ème} \text{ fois après } Y_3(T_{n-1}) = 0 \\ L\left[Y(t)\right] = 1 & \text{si } Y_3(t) \neq 0 \quad \text{ou } Y_3(t) = 0 \text{ pour la } 1^{ère} \text{ fois après } Y_3(T_{n-1}) = 0 \end{cases}$$

Par ailleurs, le mouvement résultant d'une légère perturbation d'un mouvement périodique, on choisira des paramètres $\{\beta_n\}$ de redimensionnement du temps constants et égaux à 1. En procédant au changement de variables $(ChVar)$, le problème devient alors équivalent à la suite de problèmes de tir redimensionnés suivants :

$$\begin{cases} \frac{dZ_n}{ds} = G_n(Z_n),\ s > 0, \\ Z_n(0) = 0, \\ L_z\left[Z_n(s_n)\right] = 0 \quad \text{et} \quad \forall s \in (0, s_n),\ \ L_z\left[Z_n(s)\right] \neq 0, \end{cases}$$

où $G_n(Z_n) = G_{n,K}(Z_n) + G_{n,P}(Z_n)$, avec :

$$G_{n,K}(Z_n) = D_n^{-1} F_K(Y) = D_n^{-1} \begin{pmatrix} Y_4 \\ Y_5 \\ Y_6 \\ -\mu \frac{Y_1}{R^3} \\ -\mu \frac{Y_2}{R^3} \\ -\mu \frac{Y_3}{R^3} \end{pmatrix} = \begin{pmatrix} \frac{Y_{n-1,4}}{Y_{n-1,1}} \left[1 + Z_{n,4}(s)\right] \\ \frac{Y_{n-1,5}}{Y_{n-1,2}} \left[1 + Z_{n,5}(s)\right] \\ Y_{n-1,6} \left[1 + Z_{n,6}(s)\right] \\ -\mu \left(\frac{Y_{n-1,1}}{Y_{n-1,4}}\right) \frac{1+Z_{n,1}(s)}{R^3} \\ -\mu \left(\frac{Y_{n-1,2}}{Y_{n-1,5}}\right) \frac{1+Z_{n,2}(s)}{R^3} \\ -\mu \left(\frac{1}{Y_{n-1,6}}\right) \frac{Z_{n,3}(s)}{R^3} \end{pmatrix},$$

$$G_{n,P}(Z_n) = D_n^{-1} F_P(Y) = D_n^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ A\vec{f_P}\left[A^T \begin{pmatrix} Y_{n-1,1}\left[1+Z_{n,1}(s)\right] \\ Y_{n-1,2}\left[1+Z_{n,2}(s)\right] \\ Z_{n,3}(s) \end{pmatrix}\right] \end{pmatrix},$$

et $R = \sqrt{Y_1^2 + Y_2^2 + Y_3^2} = \sqrt{Y_{n-1,1}^2\left[1+Z_{n,1}(s)\right]^2 + Y_{n-1,2}^2\left[1+Z_{n,2}(s)\right]^2 + \left[Z_{n,3}(s)\right]^2}$.
La figure 6.4 illustre une similarité faible des systèmes redimensionnés dans le plan $(s, Z_{n,i})$. Cependant, aucune similarité forte n'a pu être démontrée.

Propriété des ratios faible :
Puisque la condition (*Nonzeroness*) ne s'applique pas à la condition EOS choisie, on commence par définir des vecteurs ratios généralisés : $\forall n > 0,\ \ R_n = D_n^{-1} Y_n$..
Nos essais numériques prouvent alors, dans le cas d'un mouvement en J_2, l'existence d'une propriété des ratios faible (i.e. numérique). Les tableaux 7.3, 7.4, 7.5 et 7.6 illustrent cette propriété pour quatre cas de conditions initiales.

Résultats numériques en ce cas de similarité faible :
L'algorithme RaPTI a été testé sur 12 cas de conditions initiales du problème en J_2, chaque condition étant donnée par ses 6 éléments orbitaux $(a_0, e_0, i_0, \omega_0, \Omega_0, M_0)$. Les tableaux 7.7 et 7.8 résument les résultats obtenus lors de la résolution sur 600 tranches (correspondant à 43 jours environ) et les tableaux 7.9 et 7.10 lors de la résolution sur 1500 tranches (correspondant à 108 jours environ).
Un modèle polynomial de degré 2, utilisant les 3 derniers ratios, a été adopté et a donné de bonnes prédictions.
Le nombre d'itérations varie d'un cas à l'autre mais reste petit relativement au nombre total de tranches et ce, malgré l'absence d'une similarité forte. Les speed-ups obtenus sont assez bons et sont illustrés dans les figures 7.10 et 7.11.

Conclusion

Dans cette thèse, nous proposons et expérimentons un schéma parallèle, l'algorithme RaPTI, utilisant une nouvelle approche qui est basée sur une méthode de calcul en tranches avec redimensionnement. Ce shéma nécessite toutefois l'existence d'une condition de fin de tranche qui puisse générer une grille grossière et la vérification, sur cette grille, d'une propriété des ratios. Ceci a pour effet de limiter son champ d'application.
Cependant, lorsqu'il est applicable, RaPTI a abouti à de bonnes prédictions et à une grande accélération des calculs (qui est l'objectif premier d'une parallélisation en temps). Ces résultats encourageants nous poussent à poursuivre nos recherches en vue de confirmer la pertinence de l'algorithme RaPTI et ce, en démontrant certains résultats de convergence et en délimitant clairement son champ d'application.

Table of Contents

0 Introduction **33**

I Rescaling & Similarities **45**

1 Time-Slicing, Rescaling and Similarity **47**
1.1 End-Of-Slice Condition . . . 47
1.1.1 End-Of-Slice Condition & Coarse Grid Generation . . . 47
1.1.2 Equivalent Initial Value Shooting Problems . . . 48
1.2 Rescaling . . . 49
1.2.1 Change of Variables . . . 49
1.2.2 Rescaled EOS Condition . . . 51
1.2.3 Resulting Rescaled Systems . . . 52
1.3 Similarity Properties . . . 53
1.3.1 Invariance . . . 53
1.3.2 Asymptotic Similarity . . . 54
1.3.3 Weak Similarity . . . 57
1.3.4 Uniform Similarity . . . 58

2 Selection of an EOS Condition **61**
2.1 Case of Explosive Problems . . . 62
2.1.1 General Choice . . . 62
2.1.2 Case of nonzeroness of $\{Y_n\}$ . . . 63
2.2 Case of Oscillatory solutions . . . 67
2.2.1 General Choice . . . 67
2.2.2 Alternative Choices . . . 70
2.3 Extension to Other Cases . . . 72
2.3.1 Extinctive Problems . . . 72
2.3.2 Problems with an attraction point l . . . 73

3 Case of an Algebraic Function **77**
3.1 Proof of the Existence of an EOS Condition defining a Coarse Grid . . . 78
3.2 Existence of $\{\beta_n\}$ Yielding Uniform Similarity . . . 81
3.3 Occurrence of Invariance and Asymptotic Similarity . . . 82
3.4 Linear Case . . . 84
3.4.1 Analytic Solution . . . 84
3.4.2 Behavior of the Solution and Choice of an EOS Condition . . . 86
3.4.3 Rescaling & Similarity Properties . . . 86

II RaPTI Algorithm & Applications 93

4 Ratio Property & RaPTI Algorithm 95
4.1 Similarity and Ratio Property . . . 99
4.1.1 Ratio Vectors and End-of-slice Relations . . . 99
4.1.2 Ratio Properties Deriving from Similarity Properties . . . 100
4.2 Case of Invariance . . . 102
4.3 Preliminary Sequential Analysis . . . 103
4.3.1 Reaching a Ratio Property . . . 103
4.3.2 Ratios Predictive Model . . . 104
4.3.3 Estimation of N . . . 106
4.4 Parallel Solving of the Rescaled systems . . . 109
4.4.1 Predicted Rescaled Systems . . . 109
4.4.2 Convergence Analysis . . . 109
4.5 RaPTI Algorithm . . . 112
4.5.1 Overview of Parallel Implementation . . . 112
4.5.2 Procedures for Reaching the Ratio Property . . . 116
4.5.3 Procedures for Predictions . . . 116
4.5.4 Procedures for Corrections . . . 117
4.5.5 Communication Step . . . 118
4.6 About the Numerical Integration . . . 120
4.6.1 Choice of an Initial Time-Step τ_0 . . . 120
4.6.2 Choice of a Numerical Method . . . 121
4.6.3 Reaching accurately the EOS condition . . . 121

5 Membrane Problem 123
5.1 Description of the problem . . . 124
5.2 Rescaling and Similarity Properties . . . 126
5.2.1 Choice of an EOS condition . . . 126
5.2.2 Rescaled Systems . . . 128
5.2.3 Critical Choice of $\{\beta_n\}$ . . . 129
5.2.4 Invariance and Asymptotic Similarity . . . 129
5.3 Numerical Results: Case of Invariance . . . 132
5.4 Numerical Results: Case of Asymptotic Similarity . . . 133
5.4.1 Existence of a Ratio Property . . . 133
5.4.2 Application of RaPTI . . . 136

6 Reaction-Diffusion Problem 143
6.1 Description of the problem . . . 144
6.2 Rescaling and Similarity Properties . . . 146
6.2.1 Choice of an EOS condition . . . 146
6.2.2 Rescaling . . . 147
6.2.3 Critical Choice of $\{\beta_n\}$ . . . 148
6.2.4 Similarity Properties . . . 150
6.3 Numerical Results: Nonlinear case . . . 152
6.3.1 Existence of a Ratio Property . . . 152
6.3.2 Application of RaPTI . . . 155
6.4 Numerical Results: Linear case . . . 157
6.4.1 Existence of a Ratio Property . . . 159
6.4.2 Application of RaPTI . . . 160

7 Satellite Trajectories **163**
7.1 Derivation of a Simplified Satellite Model 164
7.1.1 Differential Equations Describing the Problem 164
7.1.2 Reference Coordinate Systems 166
7.1.3 Variables of the Motion 168
7.1.4 Coordinate Transformations 170
7.1.5 Mathematical Analysis of the J_2-problem 172
7.2 J_2-perturbed Motion as a System of ODE's 178
7.2.1 Our Coordinate System: Initially Perifocal Coordinate Frame (IPQW-Frame) 178
7.2.2 Explicit Expression of $\vec{\nabla}U$ 178
7.2.3 Equivalent System of First Order ODE's 180
7.3 RaTI Applied to a Keplerian Motion 181
7.3.1 EOS Condition, Rescaling & Invariance 181
7.3.2 Numerical Results: Application of RaTI 185
7.4 RaPTI Applied to a J_2-Perturbed Motion 186
7.4.1 Choice of an EOS Condition 186
7.4.2 Rescaling 188
7.4.3 Existence of a Weak Ratio Property 190
7.4.4 Numerical Results: Application of RaPTI 194

8 Conclusion & Perspectives **201**

9 Appendix 1: Procedures used in RaPTI **205**

10 Appendix 2: Some Features of Parallel Implementation **211**

Bibliography **217**

List of Figures

1 Multiple-Shooting Methods: Gaps on the Coarse Grid 36
2 RaPTI Algorithm: Gaps on the Coarse Grid 42

1.1 Time Slicing and Rescaling (Case of Nonzeroness) 51

2.1 EOS Condition: Scalar Explosive Positive Increasing Problem 63
2.2 Oscillating Problem: Rotation Angle 67
2.3 EOS Condition: Oscillating Problem 69
2.4 EOS Condition: Scalar Second Order Periodic Problem 69
2.5 EOS Condition: Scalar Extinctive Positive Decreasing Problem 73

4.1 Preliminary Analysis, preceding RaPTI 96
4.2 RaPTI Algorithm - A New Parallel-in-Time Scheme 97
4.3 RaPTI Algorithm - Gaps Between exact and Corrected Values 109
4.4 RaPTI Algorithm - Cyclic Distribution of Time Slices 112
4.5 RaPTI Algorithm - Classical approach v/s Duplication Approach 113
4.6 RaPTI Algorithm - Organigram. 115
4.7 RaPTI Algorithm - Communication step 119

5.1 Membrane Problem - Different Behaviors, when $b = 1$ and $p, q > 1$. 124
5.2 Membrane Problem - Global Behavior, when $0 < p \leq q \leq \frac{2p}{p+1} \leq 1$. 125
5.3 Membrane Problem - EOS Condition 127
5.4 Membrane problem - 100 slices, $b = 1$, $p = 0.7$, $q = 0.77$, $Y_0 = (1, 1)$. 127
5.5 Membrane Problem - A Case of Invariance, $b = 1$, $p = 0.7$, $q = \frac{2p}{p+1}$, $Y_0 = (1, 1)$. 131
5.6 Membrane Problem - A Case of Asymptotic similarity, $b = 1$, $p = 0.7$, $q = 0.77$, $Y_0 = (1, 1)$. 131
5.7 Membrane - Cases of Invariance: Speed-up, applying RaPTI 133
5.8 Membrane Problem - Cases of Asymptotic Similarity: Efficiencies and speed-ups, $\epsilon^g_{tol} = 5 \times 10^{-6}$, $\epsilon^{n_s}_{tol} = 10^{-5}$. 137
5.9 Membrane Problem - Cases of Asymptotic Similarity: Efficiencies and speed-ups, $\epsilon^g_{tol} = 10^{-5}$, $\epsilon^{n_s}_{tol} = 10^{-4}$. 138
5.10 Membrane - Impact of the choices of ϵ^g_{tol} and $\epsilon^{n_s}_{tol}$. 139
5.11 Membrane - Comparative Speed-ups, for different ϵ^g_{tol}. 140
5.12 Membrane - Comparative Speed-ups, for different $\epsilon^{n_s}_{tol}$. 141

6.1 Reaction-Diffusion Problem - Global Behavior, when $m = 0.7$, $p = 0.9$, $a = 3$ and $u_0(x) = 1 - x^2$, with $\Omega = [-1, 1] \subset \mathbb{R}$. 144
6.2 Reaction-Diffusion Problem - Solution Profile, when $m = 0.7$, $p = 0.9$, $a = 3$ and $u_0(x) = 1 - x^2$, with $\Omega = [-1, 1] \subset \mathbb{R}$. 146

6.3 Reaction-Diffusion Problem - EOS Condition, when $m = 0.7$, $p = 0.9$, $a = 3$ and $u_0(x) = 1 - x^2$, with $\Omega = [-1, 1] \subset \mathbb{R}$. 147
6.4 Non Linear RD Problem - Z versus s, on 30 slices, when $m = 0.7$, $p = 0.9$, $a = 3$, $S = 2$, with $\Omega = [-1, 1] \subset \mathbb{R}$ and $u_0(x) = 1 - x^2$ 152
6.5 Non Linear RD Problem - Efficiencies and speed-ups. 157
6.6 Non Linear RD Problem - Speed-up Using 8 Cores 157
6.7 Linear RD Problem - Z versus s, on 30 slices, $m = p = 1$, $a = 3$, $S = 2$, with $\Omega = [-1, 1] \subset \mathbb{R}$ and $u_0(x) = 1 - x^2$. 159
6.8 Linear RD Problem - Efficiencies and speed-ups. 161

7.1 Earth Centered Inertial Coordinate Frame (ECI) 167
7.2 Perifocal Coordinate Frame (PQW) 168
7.3 Satellite Orbital Elements (a, e, E, θ) 169
7.4 Satellite Orbital Elements (i, ω, Ω) 170
7.5 Satellite - Keplerian motion ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$). 182
7.6 Satellite - EOS Condition for Keplerian Motion ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$) 183
7.7 Satellite, J2-perturbed Motion - EOS Condition and Orbit ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$ and $M_0 = 123°$) 187
7.8 Satellite, J2-perturbed Motion - Y_1 versus t ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$) 188
7.9 Satellite, J2-perturbed Motion - Z_n versus s on 80 time-slices ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$). 192
7.10 Satellite, J2-perturbed Motion - Efficiencies and Speed-ups, Averaged on the 12 Cases, for $N = 600$. 197
7.11 Satellite, J2-perturbed Motion - Efficiencies and Speed-ups, Averaged on the 12 Cases, for $N = 1500$. 199

List of Tables

5.1 Membrane Problem - Different Behaviors when $b = 1$ and $p, q > 1$. 125
5.2 Membrane - Cases of Invariance: Application of RaPTI algorithm *by solving all slices, with exact predictions*, with $b = 1$ and $q = \frac{2p}{p+1}$. 132
5.3 Membrane - Cases of Invariance: Application of RaTI algorithm *by solving only one slice*, with $b = 1$ and $q = \frac{2p}{p+1}$. 133
5.4 Membrane: Ratio Property in a Case of Asymptotic Similarity 135
5.5 Membrane Problem - Cases of Asymptotic Similarity: Application of RaPTI, when $\epsilon_{tol}^{g} = 5 \times 10^{-6}$, $\epsilon_{tol}^{n_s} = 10^{-5}$. 137
5.6 Membrane Problem - Cases of Asymptotic Similarity: Application of RaPTI, when $\epsilon_{tol}^{g} = 10^{-5}$, $\epsilon_{tol}^{n_s} = 10^{-4}$. 138

6.1 RD Problem - Different Behaviors of the Solution 144
6.2 Non Linear RD Problem - Asymptotic Ratio Property (1). 153
6.3 Non Linear RD Problem - Asymptotic Ratio Property (2). 154
6.4 Non Linear RD Problem - Asymptotic Ratio Property (3). 154
6.5 Non Linear RD Problem - Values of n_s for different $\epsilon_{tol}^{n_s}$. 155
6.6 Non Linear RD Problem - Application of RaPTI (1). 156
6.7 Non Linear RD Problem - Application of RaPTI (2). 156
6.8 Linear RD Problem - Asymptotic Ratio Property (1). 160
6.9 Linear RD Problem - Asymptotic Ratio Property (2). 160
6.10 Linear RD Problem - Application of RaPTI. 161

7.1 Satellite, Keplerian Motion - Application of RaTI (1). 186
7.2 Satellite, Keplerian Motion - Application of RaTI (2). 186
7.3 Satellite, J2-perturbed Motion - Ratio Property, (A) 193
7.4 Satellite, J2-perturbed Motion - Ratio Property, (B) 193
7.5 Satellite, J2-perturbed Motion - Ratio Property, (C) 194
7.6 Satellite, J2-perturbed Motion - Ratio Property, (D) 194
7.7 Satellite, J2-perturbed Motion - Application of RaPTI, $N = 600$ (1). 196
7.8 Satellite, J2-perturbed Motion - Application of RaPTI, $N = 600$ (2). 197
7.9 Satellite, J2-perturbed Motion - Application of RaPTI, $N = 1500$ (1). 198
7.10 Satellite, J2-perturbed Motion - Application of RaPTI, $N = 1500$ (2). 199

10.1 List of Notations 216

Chapter 0

Introduction

Consider the first order initial value problem in which one seeks $Y : [t_0, T] \to \mathbb{R}^K$, such that:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t > 0, \\ Y(t_0) = Y_0, \end{cases}$$

Assuming that the existence and uniqueness of a solution is well established on $[0, \infty)$, the goal of this thesis is to propose a parallel-in-time method for solving (S).

Parallel Methods

Recently, considerable attention has been paid to the development of efficient algorithms for the numerical solution of ODE's of initial value type of form (S), since some problems could generate a system made of many thousands of ODE's which has to be solved over a long time scale. With the advent of massively parallel computers with thousands of processors, parallelism becomes the key direction for such efficient algorithms.
Different type of parallelism can be identified in attempting to solve (S).
According to [1] and [2], the parallel methods devised so far can be roughly collected in three basic categories:

1. **Parallelism across the method:**
   The computation required to perform a single integration step of a given numerical method is split (in some way) among parallel computing units.
   Direct methods exploiting such parallelism consist in performing several function evaluations concurrently on different processors. This is possible with multi-stage methods such as Runge-Kutta methods and could be interesting when the problem size is large or function evaluations are costly.
   Indirect methods, such as predictor-corrector methods, can prove to be efficient. They are based on the concept of a block method in which a block of values is predicted concurrently by some explicit method from a previous set of computed value, which are then corrected a number of times by an implicit method using a fixed-point approach.

2. **Parallelism across the system (space):**
   The parallelism is exploited at the level of the problem to be solved, when using temporal iterative techniques: for example by defining suitable splittings of the continuous problem and corresponding Picards type iterations. If the size of the problem is K, then an iterative method generating a sequence of iterative solutions

on the region of integration allows a very natural parallelism by decoupling the problem into K independent parallel problems.

3. **Parallelism across time (i.e. across the steps):**
   In fact, there is no natural parallelism across time since the solution on a time level must be known before the computation of the solution at subsequent time levels can start. However, it could be possible to compute on many time levels simultaneously by providing the processors assigned to "later" time levels some initial guess for the solution.

Actually, efficient parallel algorithms may well take elements from all three of these categories, in order to enable a more effective use of of a higher number of processors. The availability of massively parallel computers can make the critical mesh resolution often remain far from the high capabilities of these supercomputers. Therefore, one possible approach enabling a more effective use of a higher number of processors, is to superpose parallelism in time-domain to the parallelism in the space-domain or across the method.

In this thesis, we are concerned with parallelism across time.

Parallelism Across Time

Motivation and objectives:
Time-parallelism is usually motivated by one of the following reasons:

- the problem alone cannot exploit all processors available;
- the given problem contains a small number of spatial degrees of freedom;
- time-advancing the solution needs to be in near-real-time.

In other words, parallel-in-time computations are mainly meant to tackle real-time problems or to be superposed to parallelism in the space or the method directions. Therefore, a good speed-up should be the main goal of any parallel-in-time algorithm (unlike parallelism in the space direction, or across the method, that should mainly insure good efficiency and scalability). Any gain in the resulting speed-up is welcome.

Historical overview:
Many algorithms have been proposed to solve evolution problems in a time-parallel fashion. One of the first (or maybe the first) has been suggested by Nievergelt [3] and led to *multiple shooting methods*. Variants of the method were then developed, in the nineties, by Bellen & al in [4] and [5], by Chartier & Philippe in [6] and by Erhel & al in [7]. Those methods combine coarse and fine resolutions in time, similarly to what is done in space for the domain decomposition methods. They are detailed further in this section.

In the late eighties, Saltz and Naik [8] and D.Womble [9] considered the time-parallel application of iterative methods and showed that instead of iterating until convergence over each time-step, different processors can be assigned to successive steps and then iterate simultaneously. The acceleration of such methods by means of a multigrid technique led to the class of *parabolic multigrid methods*, as introduced previously by Hackbusch in [10].

The *multigrid waveform relaxation* also belongs to that class, and a successful time-parallel variant was shown by Vandewalle & al in [11] and [12].

In 2001, Lions, Maday & Turinici proposed in [13] the *parareal algorithm*. Its main goal concerns evolution problems whose solution can not be obtained in real time using one processor only, hence the proposed terminology of "parareal".
Parareal algorithm marked a turning point in solving time-dependent differential equations in a time parallel way. Since its introduction, it has received a lot of attention and was subject for many contributions.
Maday & Bal proposed in [14] an improved version of the Parareal Algorithm which gives better answers for nonlinear problems and more importantly allows to tackle non-differentiable problems (whereas the former implementation was based on the linearization of the PDE). Farhat & Chandesris presented in [15] an original mathematical justification to the framework by using the theory of distribution and gave a variant to the Parareal: the "PITA" algorithm.
Other developments and applications are also found in [16], [17], [18], [19].
Then, many contributions, based on the former works, have been peoposed during the 15th Domain Decomposition Conference (Berlin 2003) in which a whole session was affected to the Parareal Algorithm, see [20], [21], [22], [23], [24].
Other contributions followed and can be found in [25], [26], [27], [28], [29].
In the 16^{th} Domain Decomposition Conference (New York 2005), Parareal Algorithm received also additional attention in [30], [31], [32], [33], [34], and [37].
Again, in the 17^{th} Domain Decomposition Conference (Austria 2006), parareal algorithm has been subject to more developments in [39], [40] and [41].
Afterward, more contributions can be found in [43], and [44].
We cite below the main idea of some of the latest contributions:
- In [27], [31], [35] and [36] Tromeur-Dervout & al introduced an "adaptativity" in the definition of the refinement of the time grid and the time domain splitting (in order to tackle stiff problems), and then proposed an adaptative parallel extrapolation algorithm (for very stiff problems).
- In [29], [33] and [34], Vandewalle and Gander showed that the Parareal Algorithm can be interpreted as a particular multiple shooting method with a coarse grid jacobian approximation in a finite difference way, and that PA can also cast into the parabolic multigrid framework because it can be reformulated as a two-level space-time multigrid with a special Jacobi-type smoother, with strong semi-coarsening in time, and selection and extension operators for restriction and interpolation.
- In [37], [38] and [42] Farhat & al proposed strategies intended to improve the performance of the related PITA algorithm for second-order hyperbolic systems, and to accelerate the solution of nonlinear structural dynamics problems.
- In [39], Bal & al proposed a symplectic parareal-type algorithm.

General Steps of Multiple-Shooting Methods & Parareal Algorithm

1. *Choice of a coarse grid*: The interval of integration $[T_0, T]$ is decomposed into subintervals (or time-slices) such that the boundary points $\{T_n\}$ are treated as a coarse time-grid.

2. *Prediction*: Initial guess of the starting values $\{Y_n^p\}$ of the solution at the onset of each slice.

3. *Iterative procedure*, until convergence of all slices, of the following:

   (a) *Parallel computation*, on the fine grid and within the different time-slices $[T_{n-1}, T_n]$, for solving independently the following sequence of initial value problems:
   $$\begin{cases} \frac{dY}{dt} = F(Y), & T_{n-1} < t \leq T_n, \\ Y(T_{n-1}) = Y_{n-1}^p. \end{cases}$$
   This yields an end value Y_n^c of the solution, at slice $[T_{n-1}, T_n]$.

   (b) *Evaluation of the "gaps"* between the corrected end values $\{Y_n^c\}$ of the slices and the predicted starting values $\{Y_n^p\}$ of the next slices.
   The iterations stop if all the jumps are sufficiently small. If not, the iterative process continue.

   (c) *Corrective step*, performed to update the seed values.

Figure 1 shows the gaps between the predicted and corrected value of the solution, for a given coarse grid.

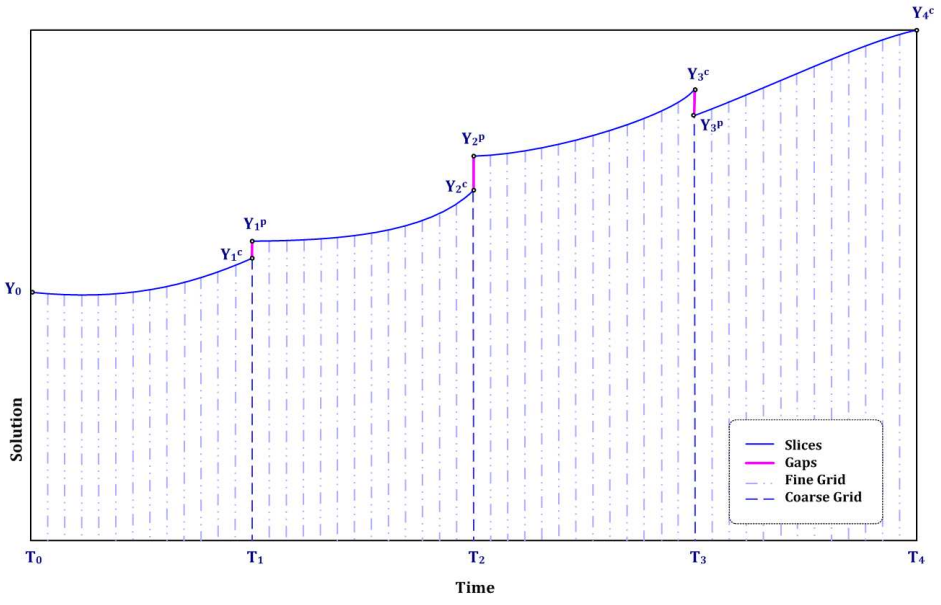


Figure 1: Multiple-Shooting Methods: Gaps on the Coarse Grid

Multiple shooting methods reduce the problem to a fixed-point problem:
Once the coarse grid of N time-slices is chosen and a sequence $\{Y_0^{(0)}, Y_1^{(0)}, Y_2^{(0)}, \cdots, Y_{N-1}^{(0)}\}$ of first predictions of the starting values at the times $\{T_0 = 0, T_1, T_2, \cdots, T_{N-1}\}$ is available, one can solve, in parallel, a sequence of N initial value problems, defined on each n^{th} time-slice $(1 \leq n \leq N)$ by:

$$\begin{cases} \frac{dY}{dt} = F(Y), & T_{n-1} < t \leq T_n, \\ Y(T_{n-1}) = Y_{n-1}, \end{cases} \tag{1}$$

For $n = 1, 2, \cdots, N$, let $\phi_n(Y_{n-1}) = Y(Y_{n-1}, T_{n-1}, T_n)$ be the value of the solution of (IVP_n), at $t = T_n$. ϕ_n is then the map:

$$\begin{array}{rccc} \phi_n : & \mathbb{R}^K & \longrightarrow & \mathbb{R}^K \\ & X & \longmapsto & \phi_n(X) = Y(X, T_{n-1}, T_n) \end{array}$$

The functions $\{\phi_n\}$ are well defined if the existence and uniqueness of a solution to (S) is guaranteed on any subinterval of $[0, T]$ and for any initial condition.
Assuming that the coarse grid is regular, one has all the ϕ_n that are equal: $\forall n, \phi_n = \phi$.
Consider now the operator:

$$\begin{array}{rccc} \Phi : & \mathbb{R}^{K\times(N+1)} & \longrightarrow & \mathbb{R}^{K\times(N+1)} \\ & U = \begin{pmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_N \end{pmatrix} & \longmapsto & \Phi(U) = \begin{pmatrix} Y_0 \\ \phi(Y_0) \\ \vdots \\ \phi(Y_{N-1}) \end{pmatrix} \end{array}$$

It is then possible to formulate problem (S) as a fixed point problem, since finding the exact solution of (S) is equivalent to finding a fixed-point U^* of Φ, solution of the equation:

$$\Phi(U) = U \Longleftrightarrow (Id - \Phi)(Y) = 0,$$

where Id is the identity operator on $\mathbb{R}^{K\times(N+1)}$.
The existence and uniqueness of such a fixed point U^* follows from the fact that $\Phi^{(N+1)}(U) = U^*$ for any U, where $\Phi^{(N+1)}(U)$ denotes the $(N+1)^{th}$ application of Φ. Hence, an iterative method, such as Newton's method, will be suitable for obtaining the fixed point of Φ.
Provided F is continuous and differentiable, Φ will be continuously differentiable on $\mathbb{R}^{K\times(N+1)}$. The k^{th} iteration of Newton's method can then be written as:

$$\left[I - J_\Phi^{(k)}\right]\left(U^{(k+1)} - U^{(k)}\right) = -U^{(k)} + \Phi\left(U^{(k)}\right),$$

where I is the identity matrix in $\mathbb{R}^{K\times(N+1)}$ and the jacobian of Φ, at the k^{th} iteration, is given by an $(N+1)\times(N+1)$ block bidiagonal matrix, with block size K, given by:

$$J_\Phi^{(k)} = \begin{pmatrix} 0 & 0 & 0 & \cdots & 0 \\ J_\phi\left(Y_0^{(k)}\right) & 0 & 0 & \cdots & 0 \\ 0 & J_\phi\left(Y_1^{(k)}\right) & 0 & \cdots & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & J_\phi\left(Y_{N-1}^{(k)}\right) & 0 \end{pmatrix}$$

This yields the following formulation, at the k^{th} iteration:

$$\forall n \in \{1, 2, \cdots, N\}, \quad Y_n^{(k+1)} = \phi\left(Y_{n-1}^{(k)}\right) + J_\phi\left(Y_{n-1}^{(k)}\right)\left(Y_{n-1}^{(k+1)} - Y_{n-1}^{(k)}\right).$$

The iterative process is repeated until k satisfies: $\left\|U^{(k+1)} - U^{(k)}\right\| \leq \epsilon$, where ϵ is the desired tolerance and $\|.\|$ is for example the euclidean norm.
Obviously, such method necessitates:
(i) an initial guess of $U^{(0)} = U = \begin{pmatrix} Y_0 \\ Y_1^{(0)} \\ \vdots \\ Y_N^{(0)} \end{pmatrix}$,
(ii) approximations of $\left\{\phi\left(Y_{n-1}^{(k)}\right)\right\}$ and $\left\{J_\phi\left(Y_n^{(k)}\right)\right\}$ (for $n = 1, 2, \cdots, N-1$), by some way or another,
(iii) a method for solving the resulting fixed-point method (such as Newton's method that is described above).

Two examples of multiple shooting methods:
Different multiple shooting methods differ from each other by the method chosen for solving the fixed-point problem and by the way the initial guess is done and the function Φ and its Jacobian are approximated.
In [6], Chartier & Philippe use Newton's method and simply start with an initial guess where for all $n = 1, 2, \cdots, N$, $Y_n^{(0)} = Y_0$. Then, they replace ϕ by an approximation $\tilde{\phi}$ computed by an ODE-solver and approximate J_ϕ by a standard finite differences approximation of $J_{\tilde{\phi}}$.
In [7], Erhel & Rault use also Newton's method and start with an initial guess obtained by using a simplified model for which an analytic solution ϕ_1 can be derived. Then, they replace ϕ by an approximation $\tilde{\phi}$ computed by an ODE-solver and approximate J_ϕ by the jacobian J_{ϕ_1} of the same analytic simplified model that has been used for the initial guess.

Parareal Algorithm:
Once a regular coarse grid is chosen (with $\Delta T = \frac{T}{N}$), the parareal algorithm, proposed by Lions et al in [13], provide the initial guess by solving problem (S) over the coarse grid using the Euler implicit first-order difference scheme, thus obtaining a sequence $\{Y_0^{(0)}, Y_1^{(0)}, Y_2^{(0)}, \cdots, Y_{N-1}^{(0)}\}$ of predicted starting values at the times $\{T_0 = 0, T_1, T_2, \cdots, T_{N-1}\}$.
Then, they solve at each k^{th} iteration, in parallel and on a very fine grid, the sequence of N initial value problems, defined on each n^{th} time-slice ($1 \leq n \leq N$) by 1, getting a solution $y_n^{(k)}(.)$ on $[T_{n-1}, T_n]$ that yields an end-of-slice value of the solution $y_n^{(k)}(T_n)$, at $t = T_n$. This define the jumps $S_n^{(k)} = y_n^{(k)}(T_n) - Y_n^{(k)}$, making the solution non continuous at $\{T_n\}$.
Iteratively, they improve the accuracy of this scheme by defining a non continuous function $\delta^{(k)}$ (corrective function) such that $y^{(k)} + \delta^{(k)}$ is a continuous function solving problem (S).
Using the same Euler implicit first-order difference scheme on the coarse grid, results is a propagation of the jumps. One gets the corrections $\delta_n^{(k)}$, at $t = T_n$, by solving:

$$\begin{cases} \frac{\delta_{n+1}^{(k)} - \delta_n^{(k)}}{\Delta T} - \delta_{n+1}^{(k)} F_Y\left(Y_{n+1}\right) = \frac{S_n^{(k)}}{\Delta T}, \\ \delta_0^{(k)} = 0, \end{cases}$$

(a linearization of F_Y should be used).
The new starting values at the next $(k+1)^{th}$ iteration are then:

$$Y_n^{(k+1)} = y_n^{(k)}(T_n) + \delta_n^{(k)}.$$

This iterative process is repeated until convergence of all slices, up to a given tolerance.

As specified by Vandewalle & al in [33], Parareal Algorithm can be interpreted as a particular multiple shooting method with a coarse grid jacobian approximation in a finite difference way.

Our New Approach for Parallelism Across Time

This thesis deals with an alternative approach, referred to as Ratio-based Parallel Time Integration Algorithm (RaPTI).
RaPTI algorithm differs from the previous paradigm: it does not start by choosing a coarse grid, rather it starts by choosing a stopping criterion (the End-Of-Slice condition) that will provide the coarse grid, and solves the problem sequentially on a few n_s time-slices.Then it performs "ratio-based" predictions (without a prior knowledge of the corresponding times!), corrects and iterates until convergence occurs on all slices.

Technique underlying RaPTI algorithm:
A rescaling methodology, introduced in [48], is the sliced-time computation that underlies RaPTI algorithm. It has been initially devised for solving explosive problems having a finite time existence of the solution.
The method starts by choosing a uniform End-Of-Slice (EOS) condition, governed by a family of continuous functions $\{E_n\}$, depending on the behavior of the solution and intended to automatically generate a coarse grid. Starting with $Y(T_{n-1}) = Y_{n-1}$ at T_{n-1}, one gets T_n by shooting the EOS condition that the solution should satisfy, which is of the form:

$$E_n[Y(T_n)] = 0 \text{ and } \forall t \in (T_{n-1}, T_n),\ \ E_n[Y(t)] \neq 0.$$

Besides, the rescaling methodology rescales, on each time-slice $[T_{n-1}, T_n]$, both the time variable t and the solution $Y(t)$ into a rescaled time variable s and a rescaled solution $Z_n(s)$ that are set to zero at the onset of every slice:

$$\begin{cases} t = T_{n-1} + \beta_n s, & \beta_n > 0, \\ Y(t) = Y_{n-1} + D_n Z_n(s), \end{cases}$$

where β_n is a time-rescaling factor and D_n is a diagonal invertible matrix, both depending solely on the starting value Y_{n-1}.
The EOS condition, together with the change of variables, makes the Initial Value Problem (S) equivalent to a sequence of rescaled Initial Value Shooting Problems in which one seeks for both the rescaled time s_n and the rescaled function $Z_n : [0, s_n] \to \mathbb{R}^K$, on each n^{th} slice corresponding to $[T_{n-1}, T_n]$, such that:

$$(S'_n) \quad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, \\ H[Z_n(s)] \neq 0,\ \forall s < s_n \text{ and } H[Z_n(s_n)] = 0, \end{cases}$$

where the functions $\{E_n\}$ governing the EOS condition are assumed to rescale into an invariant function H.

Similarity properties of the rescaled problems:
In line with the above work, it has been thought that this sliced-time computation can be extended to problems having a global time existence, in order (i) to control the accuracy and growth of the computed solution, whenever (S) exhibits stiffness properties and (ii) to devise efficient parallel-in-time algorithms to compute $Y(t)$.
Such extension starts with the definition of some similarity properties satisfied by the rescaled problems $(S'n)$:

1. *Uniform similarity*: when the boundedness of the solution Z_n implies that of $G_n(Z_n)$ and $J_{G_n}(Z_n)$. This property allows, by introducing an end-of-slice that bounds Z_n, to control the stiffness of the problem.

2. *Invariance*: when all the functions G_n are equal. This is the case of perfect similarity and yields equal end-of-slice values of the rescaled solution $\{Z_n(s_n)\}$.

3. *Asymptotic similarity*: when the functions G_n converge uniformly to a limit function G_L defining a "limit problem". This yields, under some conditions, the convergence of the end-of-slice values of the rescaled solution $\{Z_n(s_n)\}$ towards that of the limit problem.

4. *Weak similarity*: when the end-of-slice values of the rescaled solution $\{Z_n(s_n)\}$ are very close, up to a given tolerance, on a certain number of slices. This similarity is only numerically defined.

Hence, in the case of invariance, asymptotic similarity and weak similarity, one has the *invariance* (or *near-invariance*) of the end-of-slice values of the rescaled solution $Z_n(s_n)$. In this thesis, we show how those 3 properties can yield a parallel-in-time scheme.

From rescaling and similarity properties to time-parallelism:
Two major reasons justify our work:

1. Relevant choices of an EOS condition and of the time-rescaling factors $\{\beta_n\}$ can yield the invariance (or near invariance) of the rescaled problems and *provide a prediction procedure* as summarized below.
   Under the nonzeroness condition that assumes: $\forall n, \forall i : 1 \leq i \leq K$, $Y_{n,i} \neq 0$, the end-of-slice value $Z_n(s_n) \in \mathbb{R}^k$ of the rescaled solution, on the n^{th} slice, relates directly the successive starting values Y_{n-1} and Y_n of the solution:

$$Y_n = D_n \left(\mathbf{1} + Z_n(s_n)\right),$$

   where D_n is the diagonal matrix having Y_{n-1} on its main diagonal and $\mathbf{1}$ is the vector of ones in $\mathbb{R}^K$.
   By letting:

$$R_n = \mathbf{1} + Z_n(s_n),$$

   and D_{R_n} be the diagonal matrix having R_n on its main diagonal, the previous relation becomes $Y_n = D_n R_n$, which is simply a component wise multiplication of the vectors R_n and Y_{n-1}, and is equivalent to:

$$Y_n = D_{R_n} Y_{n-1}.$$

This shows how R_n can be viewed as a "ratio-vector" between the consecutive starting values Y_{n-1} and Y_n of the solution.
The cases of invariance or near-invariance of the rescaled problems, usually yield a quasi-stabilization of the ratios: a *ratio property* is said to be detected. This is the starting point of our approach, since it makes the end-of-slice values of the rescaled solution $\{Z_n(s_n)\}$, and therefore the ratio-vectors R_n and the starting values of the solution $\{Y(T_n)\}$ at the beginning of each time-slice, well-suited to be predicted through the so-called *ratio-based prediction procedure*.
The main idea of this prediction procedure, is to start with some sequential computations on a relatively small number n_s of time-slices, until reaching numerically a ratio property, and then to use the resulting n_s exact ratio-vectors for predicting the next ones. A backward analysis is then done on the exact ratios and a mathematical model approximating the sequence $\{R_n\}$ $(n \leq n_s)$, as a function of n, is chosen. This model can be used for predicting (by extrapolation) the next ratios $\{R_n^p\}_{n>n_s}$ and therefore the next starting values $\{Y_n^p\}$, using the exact sequentially computed $Y_{n_s}^e$ and the predicted ratios:

$$Y_n^p = D_{R_n^p} \cdots D_{R_{n_s+2}^p} D_{R_{n_s+1}^p} Y_{n_s}^e.$$

Moreover, and in case of asymptotic similarity, the mathematical model could take into consideration the convergence of $\{R_n\}$ toward R_L that can be evaluated by solving a unique slice.
Such prediction procedure is repeated at the beginning of each iteration, after updating n_s by the number of the last converged slice at the previous iteration.

2. By setting the rescaled time s to 0 at the beginning of each slice, rescaling allows, once the predictions are available, parallel computations of the rescaled systems (S'_n) to be done within a *local time*, since the sequence $\{T_n\}$ is a priori unknown.

Preliminary analysis:
Thus, one should first reach numerically a ratio property, by solving sequentially a certain number n_s of time-slices.
Afterwards, a backward analysis on the obtained exact ratios should be done and a mathematical model approximating the sequence $\{R_n\}$ $(n \leq n_s)$, should be chosen.
At this point only, the application of RaPTI algorithm can be considered.

Description of RaPTI Algorithm:

General steps of RaPTI Algorithm:

1. *Sequential run on n_s slices*, until reaching a ratio property (n_s should be much less than the total number of slices).

2. *Iterative procedure*, until convergence of all slices, of the following:

   (a) *Prediction*: Initial guess of the starting values $\{Y_n^p\}$ (for $n > n_s$) of the solution at the onset of each slice, using the mathematical model fitting the last computed ratios.

   (b) *Parallel computation*, on the fine grid and within the different time-slices $[0, s_n]$, for getting both s_n and $Z_n(.)$ by solving independently the following initial value shooting problems:

$$(S'_n) \quad \begin{cases} \frac{dZ_n}{ds} = G_n^p(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, \\ H\left[Z_n(s)\right] \neq 0, \ \forall s < s_n \ \text{ and } \ H\left[Z_n(s_n)\right] = 0, \end{cases}$$

   This yields an end value Y_n^c of the solution, at slice $[T_{n-1}, T_n]$.
   After having solved an n^{th} slice, each processor *evaluates the gap* between Y_n^c and Y_n^p and *keeps solving as long as the gaps are sufficiently small*. The iterations stop if all the gaps are sufficiently small.
   If not, the iterative process continue, after updating n_s by the number of (so far) converged slices.

Figure 2 shows the gaps between the predicted values Y_n^p and the corrected value Y_n^c of the solution.

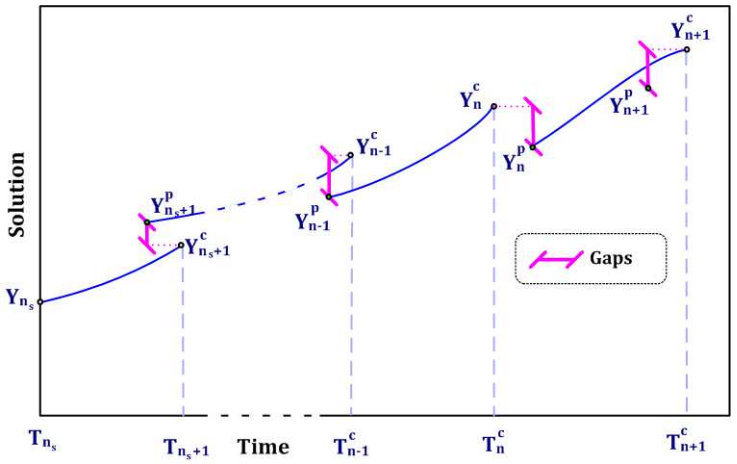


Figure 2: RaPTI Algorithm: Gaps on the Coarse Grid

Note that Y_n^p and Y_n^c do not correspond exactly to the same time, and that the time corresponding to Y_n^p is not predicted and not even considered. However, it will be proved, under some assumptions, that if the predictions are good enough and Y_n^p and Y_n^c are close enough, then both Y_n^c and T_n^c converge to their exact values.
Note also that the fine grid cannot be represented in figure 2, even if it is regular, since it applies on the rescaled time and is therefore modulated by the time-rescaling factors $\{\beta_n\}$. Moreover, this fine grid is refined at each end-of-slice, in order to reach accurately the EOS condition.

Plan of the Thesis

This thesis is globally subdivided in two parts: Part I deals with the rescaling methodology and the resulting similarity properties on which our new approach to time-parallelism is based. Part II describes the parallel-in-time scheme we are proposing (RaPTI algorithm) and provides numerical applications on three problems.

Part I: Rescaling & Similarity:
In *Chapter 1*, we describe our automatic generation of a coarse grid, based on a relevant end-of-slice condition, that makes solving the initial value problem (S) equivalent to solving an infinite sequence of initial value shooting problems (S_n). On each time-slice, one would seek both the time T_n and the solution $Y : [T_{n-1}, T_n] \to \mathbb{R}^K$, thus getting progressively the coarse grid. Then, we rescale, on each time-slice $[T_{n-1}, T_n]$, the time-variable t and the solution $Y(.)$ into a rescaled time s and a rescaled solution $Z_n(.)$, in a way to set them at 0 at the onset of every slice. The problem becomes equivalent to an infinite sequence of rescaled initial value shooting problems (S'_n), in which one seeks both the rescaled time s_n and the rescaled solution $Z_n : [0, s_n] \to \mathbb{R}^K$.
Three types of similarity properties are mathematically defined on the rescaled systems: uniform similarity, asymptotic similarity and invariance. Under some conditions, we prove that the simultaneity of uniform and asymptotic similarity yields the convergence of the solution of problems (S'_n) towards that of a limit problem, as n goes to infinity. Besides, a weak similarity is also numerically defined.

Since choosing a relevant EOS condition is at the core of our technique, *Chapter 2* is devoted to the selection of this condition. Knowing that such selection is problem dependent and requires a prior knowledge of the global behavior of the solution, we give some ways for selecting an EOS condition in four cases of behavior: (i) explosive solution, (ii) extinctive solution, (iii) solutions having an attraction point and (iv) oscillatory solution. Note that the numerical experiments of this thesis deal only with the explosive and oscillatory cases.

Chapter 3 is mainly of theoretical interest: it deals with the case where the function F defining the initial value problem (S) has the algebraic form component-wise given by $(F(Y))_i = \sum_j a_{ij}(Y_i)^{k_{ij}}(Y_j)^{l_{ij}}$. After proving the existence of an EOS condition generating a coarse grid, we propose a critical choice for the time-rescaling parameter β_n yielding a uniform similarity and the boundedness of $\{s_n\}$, independently of n. Then, the particular linear case is considered and an asymptotic similarity property proved, under some conditions.

Part II: RaPTI Algorithm & Applications:
In *Chapter 4*, we start showing how the rescaling methodology, with the various types of similarity properties (invariance or near-invariance), could yield corresponding types of ratio properties. After considering the invariant case that results in a rare perfect parallel algorithm, we describe the preliminary sequential analysis that is necessary, in the general case: it allows the detection of a ratio property and provides a mathematical model, fitting the ratios, that will be used in the prediction procedure. Then, we show how the rescaled systems are solved in parallel, using predicted functions $\{G_n\}$ and getting corrected end-of-slice values. We also provide a "convergence test" that should hold in order to guarantee, under some assumptions, the convergence of the corrected solution toward

the exact one. Afterwards, RaPTI algorithm is detailed in all of its steps and its numerical implementation is discussed.

RaPTI algorithm proved to be efficient in [53], when tested on Reaction-Diffusion problems with bounded solutions and Lotka-Volterra models for 2 and 3 species since these problems revealed a ratio property. In this thesis, the application of RaPTI is extended to problems having an explosive or oscillatory behaviors of their solution.
In *Chapter 5*, we consider a membrane problem, that is a second order scalar initial value problem:

$$y^{''} - b|y^{'}|^{q-1}\ y^{'}\ +\ |y|^{p-1}\ y = 0,\ \ t > 0, \quad y(0) = y_{1,0}, \quad y^{'}(0) = y_{2,0},$$

when the combination of problems parameters (p, q) yield an explosive behavior on $[0, \infty)$, in an oscillatory way. In the case where $q = \frac{2p}{p+1}$, the rescaling technique yields a case of invariance, and therefore a perfect parallelism. Whereas in case $0 < p \leq q < \frac{2p}{p+1}$, we obtain an asymptotic similarity and the application of RaPTI algorithm yields excellent speed-ups.

In *Chapter 6*, RaPTI algorithm is applied to the reaction-diffusion problem of the form:

$$\frac{\partial u}{\partial t} - \Delta u^m = au^p,\ x \in \Omega \subset \mathbb{R}, \quad u(x,t) = 0,\ x \in \partial\Omega\ \ u(x,0) = u_0(x) > 0,$$

where $a > 0$ and $0 < m \leq p \leq 1$. For this combination of problems parameters (m, p), the solution exhibits, on $[0, \infty)$, an explosive behavior. The application of the rescaling methodology yields an asymptotic ratio property, allowing a quick convergence of RaPTI algorithm and a good speed-up. The particular linear case $m = p = 1$ is also discussed.

Chapter 7 deals with the satellite trajectories that are computed by solving a system of second order differential equations that follows from the general equation of motion given by Newton's second law and has the form:

$$\ddot{\vec{r}}(t) = \vec{f}\,(\vec{r}\,),\ \ t > 0, \quad \vec{r}(0) = \vec{r_0}, \quad \dot{\vec{r}}(0) = \dot{\vec{r}}_0,$$

where $\vec{r}$ and $\dot{\vec{r}}$ are respectively the position and velocity vector of the satellite, and $\vec{f}\,(\vec{r})$ is the resulting force applied to the satellite.
When $\vec{f}\,(\vec{r})$ reduces to the centered gravitational attraction of the earth, the motion is Keplerian, i.e. the satellite moves in a fixed plane through the center of the earth, along an ellipse having this center as one focus. When the satellite is subject to perturbing forces, we may still consider the satellite orbit as an ellipse but then, the parameters of this ellipse (the orbital elements) will no longer be constant: at each instant, this ellipse will be slightly different.
In a first part of the chapter, we derive a simplified satellite problem: the J_2-perturbed problem.
In the case of a Keplerian motion, the rescaling technique yields a case of invariance, and therefore a perfect parallelism. In a J_2-perturbed problem, we obtain a weak similarity and the application of RaPTI algorithm yields significant speed-ups.

Part I

Rescaling & Similarities

Chapter 1

Time-Slicing, Rescaling and Similarity

We consider the first order initial value problems in which one seeks the solution $Y : [0, \infty) \to \mathbb{R}^K$, such that:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t > 0 \\ Y(0) = Y_0. \end{cases}$$

In line with some "rescaling" adaptative methods, introduced by Chorin in [45] and carried on in [46], [47], our approach to the numerical integration of (S) is a *two-level* procedure, devised by Nassif & al in [48] and developed in [49], [50] and [51]. It consists in:

1. Generating a coarse grid of time slices $\{[T_{n-1}, T_n]\,,\ n \geq 1\}$, starting at $T_0 = 0$, on the assumption of the existence of an End-Of-Slice condition determined by the behavior of the solution (Rf assumption 1).

2. Solving (S) on each of the slices $[T_{n-1}, T_n]$, after rescaling the time variable and the solution. Those rescaled systems may have some similarity properties yielding a relationship between $Y(T_{n-1})$ and $Y(T_n)$.

1.1 End-Of-Slice Condition

The generation of a coarse grid $\{[T_{n-1}, T_n]\,,\ n \geq 1\}$ is based, in general, on the behavior of the solution $Y(t)$ that provides a uniform rule for ending a slice, in a way to make (S) equivalent to an infinite sequence of initial value shooting problems. Such rule is called an End-Of-Slice condition and is the core of the present procedure.
It differs from the traditional sliced-time approach of selecting equally-spaced time intervals, whereby $T_{n-1} - T_n = \Delta T =$ constant $(\forall n)$. It rather uses a uniform stopping criterion that the end-of-slice value of the solution should satisfy. Hence, one gets T_n from T_{n-1}, for all $n \geq 1$, by shooting to the corresponding End-Of-Slice condition.

1.1.1 End-Of-Slice Condition & Coarse Grid Generation

The present time-slicing procedure consists in generating a sequence of time slices $[T_{n-1}, T_n]$, called coarse grid, such that:

$$\cup_{n \geq 1} [T_{n-1}, T_n] = [0, \infty), \tag{1.1}$$

on the basis of an End-Of-Slice (EOS) condition.

In all what follows, "End-Of-Slice condition" will be referred to as "EOS condition".

This EOS condition is usually governed by a family of continuous functions $\left\{E_n : \mathbb{R}^K \longrightarrow \mathbb{R}\right\}$ that strongly relates the starting and ending values of the solution on each n^{th} time-slice.
Typically, each function E_n, associated with an initial value problem:

$$(IVP_n) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t \geq T_{n-1} \\ Y(T_{n-1}) = Y_{n-1}, \end{cases}$$

is parametrized by the the slice index n, through the starting value $Y_{n-1} = Y\left(T_{n-1}\right)$, and allows to end a time-slice, *in a unique way*, when $E_n\left[Y(t)\right] = 0$ and such that for all n:

$$\begin{array}{ll} \text{at } t = T_n, & E_n\left[Y\left(T_n\right)\right] = 0, \\ \forall t \in \left(T_{n-1}, T_n\right), & E_n\left[Y\left(t\right)\right] \neq 0. \end{array} \tag{1.2}$$

The selection of the family of functions $\{E_n\}$ should be such that the EOS condition (1.2) is guaranteed to be met infinitely many times, thus providing a unique increasing sequence $\{T_n\}_{n\geq 0}$ satisfying (1.1). Hence, such selection is problem-dependent and requires a prior knowledge of the behavior of the solution.
Usually, the resulting coarse grid has non-constant time slices.
Throughout this thesis, the following assumption is considered to be satisfied.

Assumption 1 ***: Existence of an EOS-defined Coarse Grid***
Given a solution $Y(t)$ of (S), there exists a specific End-Of-Slice (EOS) condition, governed by a family of functions $\{E_n\}$ and yielding, in a unique way, an increasing sequence $\{T_n\}_{n\geq 0}$ satisfying (1.1): $\cup_{n\geq 1}[T_{n-1}, T_n] = [0, T]_{T\leq\infty}$.

1.1.2 Equivalent Initial Value Shooting Problems

The generation of a coarse grid transforms the Initial Value Problem (S) into an equivalent sequence $\{(S_n)\}$ of *Initial Value Shooting Problems* in which one seeks, on each n^{th} slice $[T_{n-1}, T_n]$, the EOS time T_n and the function $Y : [T_{n-1}, T_n] \to \mathbb{R}^K$ such that:

$$(S_n) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & T_{n-1} < t \leq T_n \\ Y(T_{n-1}) = Y_{n-1}, & \\ E_n\left[Y(T_n)\right] = 0 \text{ and } \forall t \in (T_{n-1}, T_n), \;\; E_n\left[Y(t)\right] \neq 0. & \end{cases}$$

The existence and uniqueness of a solution to each of those initial value shooting problem (S_n), i.e. the existence and uniqueness of a time T_n and a solution $Y : [T_{n-1}, T_n] \to \mathbb{R}^K$ solving (S_n), follow from that of the original problem (S) and from assumption 1.

Example:
An "academic" example could be the scalar initial value problem in which one seeks for $Y : \mathbb{R} \to \mathbb{R}$ such that:

$$\frac{dY}{dt} = aY^p \;\; (a > 0 \;\; p > 0) \;, \quad Y(0) = Y_0 > 0, \tag{1.3}$$

The positive solution Y is analytically given by $Y(t) = \left[Y_0^{1-p} + (1-p)t\right]^{\frac{1}{1-p}}$, which monotonously grows to infinity (in infinite time if $p > 1$, in finite time if $0 < p < 1$). Knowing that $Y(t) \neq 0$ $(\forall t)$, implying $Y_n = Y(T_n) \neq 0$ $(\forall n)$, a possible choice of functions $\{E_n\}$ governing the EOS condition would be, in this case:

$$E_n\left[Y(t)\right] = \frac{Y(t) - Y_{n-1}}{Y_{n-1}} - S \tag{1.4}$$

S being a positive cut-off value that the relative variation of the solution should not exceed. Because of the explosive behavior of the solution, this EOS condition is guaranteed to be reached infinitely many times. Moreover, one has specifically:

$$\begin{cases} \text{at } t = T_n, & E_n\left(Y(t)\right) = 0, \\ \forall t \in (T_{n-1}, T_n), & E_n\left(Y(t)\right) < 0, \\ \forall t > T_n, & E_n\left(Y(t)\right) > 0. \end{cases}$$

1.2 Rescaling

The previously described time-slicing technique makes the Initial Value Problem (S) equivalent to a sequence (S_n) of Initial Value Shooting Problems. Dealing uniformly with $\{(S_n)\}$ is then done through a rescaling technique that rescales, on each time-slice $[T_{n-1}, T_n]$, both the time variable t and the solution $Y(t)$ into a rescaled time variable s and a rescaled solution $Z_n(s)$. The sequence $\{(S_n)\}$ of initial value shooting problems is, in turn, equivalent to a sequence $\{(S'_n)\}$ of *rescaled* initial value *shooting* problems.

1.2.1 Change of Variables

On every slice $[T_{n-1}, T_n]$, the time t and the solution Y are changed into a rescaled time s and a rescaled solution Z_n, such that:

$$\begin{cases} t = T_{n-1} + \beta_n s, \qquad \beta_n > 0 & (1.5.1) \\ Y(t) = Y_{n-1} + D_n Z_n(s), & (1.5.2) \end{cases} \tag{1.5}$$

where:

- $Y_{n-1} = Y(T_{n-1}) \in \mathbb{R}^K$
- $D_n = \mathrm{diag}(\alpha_n) \in \mathbb{R}^{K \times K}$ is a diagonal matrix associated with the vector $\alpha_n \in \mathbb{R}^K$ that is defined in (1.6), in terms of the initial value of the solution Y_{n-1} as follows:

$$\alpha_{n,i} = \begin{cases} Y_{n-1,i} & \text{if } Y_{n-1,i} \neq 0 \\ 1 & \text{if } Y_{n-1,i} = 0 \end{cases} \tag{1.6}$$

- $\beta_n > 0$ is a time-rescaling factor depending solely, and continuously, on the starting value Y_{n-1} and chosen so that the solution can be controlled. The rescaled time-steps are constant on all slices, but are refined at the end of the slices, in order to reach accurately the EOS condition. It should be noted that β_n modulates the constant rescaled time-steps (in s), generating adaptive real time-steps (in t).

Thus, on each slice, rescaling depends on the chosen time-rescaling factor β_n.

Note that the choice (1.6) for α_n makes the diagonal matrix D_n invertible and yields on each slice n:

$$Z_n(s) = D_n^{-1}(Y(t) - Y_{n-1}),$$

or more explicitly, for all $i \in \{1, 2, ..., K\}$:

$$Z_{n,i}(s) = \begin{cases} \frac{Y_i(t) - Y_{n-1,i}}{Y_{n-1,i}} & \text{if } Y_{n-1,i} \neq 0, \\ Y_i(t) & \text{if } Y_{n-1,i} = 0. \end{cases}$$

Note also that the rescaled time $s = \frac{t - T_{n-1}}{\beta_n}$ and the rescaled solution $Z_n(s)$ are set to 0 at the beginning of every slice. This allows each of the initial value shooting problems (S_n) to be solved through a local approach.
Besides, $s_n = \frac{T_n - T_{n-1}}{\beta_n}$ is the end-of-slice rescaled time corresponding to T_n and, since $s = 0$ at the beginning of the n^{th} slice, s_n represents the size of the n^{th} rescaled time-slice.

End-of-slice Invariances:
One should notice the dependence of the solution function Z_n on β_n, on each n^{th} slice, in the sense that *different choices of β_n lead to different functions Z_n*. However, one has the following end-of-slice invariances, independently of β_n.

- From the change of variable (1.5.1), one deduces at the end of the n^{th} slice:

$$\forall \beta_n, \quad T_n = T_{n-1} + \beta_n s_n, \tag{1.7}$$

  yielding:

$$\forall \beta_n, \quad \Delta T_n = T_n - T_{n-1} = \beta_n s_n. \tag{1.8}$$

  i.e. *the product $\beta_n s_n$, that is equal to the size ΔT_n of the n^{th} slice, is independent of the choice of β_n.*

- From the change of variable (1.5.2) one deduces at the end of the n^{th} slice:

$$\forall \beta_n, \quad Y_n = Y_{n-1} + D_n Z_n(s_n), \tag{1.9}$$

  yielding:

$$\forall \beta_n, \quad Z_n(s_n) = D_n^{-1}(Y_n - Y_{n-1}). \tag{1.10}$$

  i.e. *the end-of-slice values $Z_n(s_n)$ of the rescaled solution are independent of the choice of β_n.*

Case of Nonzeroness:
In general, we choose the EOS condition in a way to satisfy the non-zeroness condition:

$$\forall n, \forall i : 1 \leq i \leq K, \, Y_{n,i} \neq 0, \tag{1.11}$$

making $\alpha_n = Y_{n-1}$ $(\forall n)$. The change of variables is, in this case:

$$\begin{cases} t = T_{n-1} + \beta_n s, \qquad \beta_n > 0 & (1.12.1) \\ Y(t) = D_n \left(\mathbf{1} + Z_n(s) \right), & (1.12.2) \end{cases} \tag{1.12}$$

where D_n is simply the diagonal matrix having Y_{n-1} on its main diagonal and $\mathbf{1}$ is the vector of ones in $\mathbb{R}^K$, yielding:

$$\forall n, \ \forall i, \quad Z_{n,i}(s) = \frac{Y_i(t) - Y_{n-1,i}}{Y_{n-1,i}}.$$

In other words, the function $Z_n(.)$ represents (under the nonzeroness condition) the relative variation of $Y(.)$ with respect to Y_{n-1} within the n^{th} slice $[T_{n-1}, T_n]$.

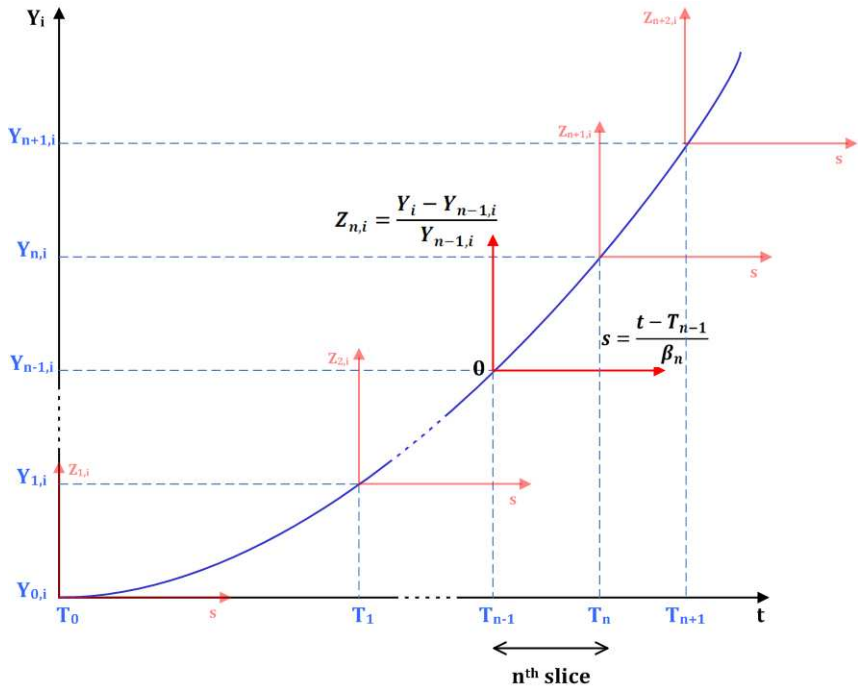


Figure 1.1: Time Slicing and Rescaling (Case of Nonzeroness)

However, the general choice (1.6) for α_n allows to tackle cases where some components of $\{Y_n\}$ could be zero.

1.2.2 Rescaled EOS Condition

$Z_n(s_n)$ denotes the end-of-slice value of the rescaled solution, at $s = s_n$.
The EOS condition (1.2), governed by the family of functions $\left\{E_n : \mathbb{R}^K \longrightarrow \mathbb{R}\right\}$, translates, after the change of variables (1.5), into:

$$\begin{array}{ll} \text{at } s = s_n, & E_n\left[Y_{n-1} + D_n Z_n(s_n)\right] = 0, \\ \forall s \in (0, s_n), & E_n\left[Y_{n-1} + D_n Z_n(s)\right] \neq 0. \end{array}$$

This defines a family of continuous functions $\{H_n\}$, derived from E_n (after rescaling) and parametrized also by the starting value Y_{n-1} of each n^{th} slice, that allows to get s_n, for all n, through the rescaled EOS condition:

$$\begin{array}{ll} \text{at } s = s_n, & H_n\left[Z_n(s_n)\right] = 0, \\ \forall s \in (0, s_n), & H_n\left[Z_n(s)\right] \neq 0. \end{array} \tag{1.13}$$

Definition 1 ***: Invariance of*** $\{H_n\}$
The parametrized family of functions $\{H_n\}$ *governing the rescaled EOS condition is said to be invariant if it is independent of* n*:*

$$\forall n,\ \forall W \in \mathbb{R}^K, \quad H_n(W) = H(W). \tag{1.14}$$

Such function H *allows then to get* s_n*, at each* n^{th} *slice, through the condition:*

$$\begin{array}{ll} \text{at } s = s_n, & H\left[Z_n(s_n)\right] = 0, \\ \forall s \in (0, s_n)\,, & H\left[Z_n(s)\right] \neq 0. \end{array} \tag{1.15}$$

Example:
In the academic scalar example (1.3), considered in the previous section, the function E_n governing the EOS condition, under the nonzeroness condition, is given by (1.4) and simplifies after rescaling to:

$$E_n\left[Y_{n-1}\left(1 + Z_n(s)\right)\right] = Z_n(s)\ -\ S.$$

Hence, a unique function $H : \mathbb{R}^K \to \mathbb{R}$ governs the rescaled EOS condition and is defined, independently of n, by:

$$\forall W \in \mathbb{R},\ H(W) = W - S. \tag{1.16}$$

Therefore, and for a given cutoff value $S > 0$, the n^{th} slice is ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{array}{lll} \text{at } s = s_n, & H\left[Z_n(s_n)\right] = 0 \iff & Z_n(s_n) = S, \\ \text{with} & Z_n(s) < S, & \text{if } 0 < s < s_n. \end{array} \tag{1.17}$$

Moreover, and because of the monotonous and explosive behavior of the solution, one has: $\forall s > s_n,\ Z_n(s) > S$.

Remark 1 *Throughout this thesis, we deal only with EOS conditions governed by functions* $\{E_n\}$ *that rescale to invariant functions* $\{H_n\}$*, i.e.* $\forall n,\ H_n = H$

1.2.3 Resulting Rescaled Systems

The Initial Value Problem (S) is now equivalent to a sequence of rescaled Initial Value Shooting Problems in which one seeks for both the rescaled time s_n and the rescaled function $Z_n : [0, s_n] \to \mathbb{R}^K$, on each n^{th} slice corresponding to $[T_{n-1}, T_n]$, such that:

$$(S'_n) \quad \left\{ \begin{array}{lll} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, & (1.18.1) \\ Z_n(0) = 0, & & (1.18.2) \\ H\left[Z_n(s)\right] \neq 0,\ \forall s < s_n \text{ and } H\left[Z_n(s_n)\right] = 0, & & (1.18.3) \end{array} \right. \tag{1.18}$$

where

$$G_n(Z_n) = \beta_n D_n^{-1} F\big(Y_{n-1} + D_n Z_n\big).$$

From the existence and uniqueness of a solution to each of the initial value shooting problem (S_n), one deduces the existence and uniqueness of a solution to each of the rescaled initial value shooting problem (S'_n), i.e. the existence and uniqueness of a time s_n and a solution $Z_n : [0, s_n] \to \mathbb{R}^K$ satisfying (S'_n).

1.3 Similarity Properties

The main goal of this section is to provide some criteria for selecting the time rescaling factors $\{\beta_n\}$.
A first criterion is the ability of $\{\beta_n\}$ to insure a "uniform similarity" property of the rescaled problems, allowing to tackle stiff problems by yielding numerical computations having the same order of magnitude on all time-slices, as done in [48], [49] and [51].
A second criterion would be to improve upon this uniform similarity, getting (i) an "invariance" property, i.e equivalent rescaled problems , which is the ideal case of similarity, or (ii) an "asymptotic similarity" to a limit problem, or (iii) a "weak similarity" up to ϵ. This criterion is intended to provide a prediction procedure when devising a parallel-in-time algorithm (as described in chapter 4).

1.3.1 Invariance

Definition 2 ***: Invariance***
The rescaled systems (S'_n) are said to be invariant if the functions $\{H_n\}$ governing the rescaled EOS condition are invariant and if the time-rescaling parameters $\{\beta_n\}$ are such that:

$$\forall n, \quad G_n(.) = G_1(.). \tag{1.19}$$

Property 1 *: Since all the functions G_n are equal to G_1, then the rescaled problems (S'_n) are all equivalent to the Initial Value Shooting Problem (S'_1), implying:*

$$\forall n, \quad Z_n(.) = Z_1(.) \qquad \textit{and} \qquad s_n = s_1. \tag{1.20}$$

Invariance is the ideal case of similarity: solving the rescaled system on one unique time-slice allows getting, through a simple change of variables, the solution on all time-slices. However, in general, rescaling does not lead to invariance.

Example:
A typical case of invariance is the scalar academic case, given in (1.3).
For a given cut-off value S, the EOS condition given in (1.4) yields, after rescaling, invariant functions $\{H_n\}$ governing the rescaled EOS condition (1.17), and makes the initial value problem (1.3) equivalent to the following sequence of rescaled initial value shooting problems:

$$(S'_n) \quad \begin{cases} \frac{dZ}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, & \\ \forall s < s_n,\ Z_n(s) < S, \text{ and } Z_n(s_n) = S. & \end{cases} \tag{1.21}$$

with $G_n(Z_n) = \beta_n a Y_{n-1}^{p-1}(1 + Z_n)^p$.
A critical choice of the sequence $\{\beta_n\}$ would then be:

$$\beta_n = \frac{1}{a}(Y_{n-1}^{1-p}), \tag{1.22}$$

yielding $\forall n,\ G_n(Z_n) = (1 + Z_n)^p = G_1(Z_1)$ and making the rescaled systems invariant, thus having exactly the same solution $\{s_n, Z_n(.)\}$ on all time-slices.

Interesting cases of invariance are met in two application problems considered in this

thesis, namely in (i) a membrane problem (see Chapter 5) and in the determination of a satellite trajectory in a Keplerian motion (see Chapter 7).

1.3.2 Asymptotic Similarity

Definition 3 ***: Asymptotic Similarity***
The rescaled systems (S'_n) are said to be asymptotically similar to a limit system if the functions $\{H_n\}$ governing the rescaled EOS condition are invariant and if the time-rescaling parameters $\{\beta_n\}$ are such that the sequence of functions $\{G_n\}$ uniformly converges to some function G_L, i.e.:

$$\forall \rho > 0, \quad \lim_{n\to\infty} \left[\max_{W \in B_\rho} \left\| G_n(W) - G_L(W) \right\|_\infty \right] = 0, \tag{1.23}$$

where:

$$B_\rho = \left\{ W \in \mathbb{R}^K, \ \|W\|_\infty \leq \rho \right\}. \tag{1.24}$$

$G_L : \mathbb{R}^K \longrightarrow \mathbb{R}^K$ defines then a "limit problem" in which one seeks for both the rescaled time s_L and the rescaled function $Z_L : [0, s_L] \to \mathbb{R}^K$, such that:

$$(S_L) \quad \left\{ \begin{array}{ll} \frac{dZ_L}{ds} = G_L(Z_L), \ 0 < s \leq s_L, & (1.25.1) \\ Z_L(0) = 0, & (1.25.2) \\ H\left[Z_L(s)\right] \neq 0, \ \forall s < s_L \ \text{ and } \ H\left[Z_L(s_L)\right] = 0. & (1.25.2) \end{array} \right. \tag{1.25}$$

Example:
A typical case of asymptotic similarity, is obtained when rescaling the scalar initial value problem:

$$\frac{dY}{dt} = aY^p + bY^q, \ (a > 0, \ b > 0, \ p > q > 0), \ Y_0 > 0. \tag{1.26}$$

The positive solution Y being, in this case also, explosive, the same choices (1.4) and (1.22) for the EOS condition and $\{\beta_n\}$ yield the following sequence of initial value shooting problems:

$$(S'_n) \quad \left\{ \begin{array}{l} \frac{dZ}{ds} = G_n(Z_n) = (1 + Z_n)^p + \gamma_n (1 + Z_n)^q, \quad 0 < s \leq s_n, \\ Z_n(0) = 0, \\ \forall s < s_n, \ Z_n(s) < S, \ \text{and } Z_n(s_n) = S. \end{array} \right. \tag{1.27}$$

where $\gamma_n = \frac{b}{a} Y_{n-1}^{q-p}$.
Obviously: $\lim_{n\to\infty} \gamma_n = 0$. It follows that: $\lim_{n\to\infty} \left[\max_{W \in B_S} G_n(W) - (1 + W)^p\right] = 0$, where $B_S = \{W \in \mathbb{R}, \ |W| \leq S\}$.
Hence, the uniform convergence of the sequence of functions $\{G_n\}$ to the function G_L given by: $G_L(Z_L) = (1 + Z_L)^p$ and therefore the asymptotic similarity of the rescaled systems (S'_n) to a limit system given by:

$$\frac{dZ_L}{ds} = G_L(Z_L) = (1 + Z_L)^p, \ (0 < s \leq s_L), \quad Z_L(0) = 0, \quad Z_L(s_L) = S. \tag{1.28}$$

Interesting cases of asymptotic similarity are met in two application problems considered in this thesis, namely in (i) a membrane problem (see Chapter 5) and in a reaction diffusion problem (see Chapter 6).

In case of asymptotic similarity, the following intends to prove, under some assumptions,

the convergence of the solution $\{s_n, Z_n(.)\}$, of successive rescaled systems (S'_n), toward the solution $\{s_L, Z_L(.)\}$ of the limit problem (S_L).

Theorem 1 : ***Convergence of*** $\mathbf{Z_n}(.)$
If:

1. *the rescaled initial value shooting problems:*

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, & \\ H\left[Z_n(s)\right] \neq 0, \ \forall s < s_n \text{ and} & H_n\left[Z_n(s_n)\right] = 0, \end{cases}$$

are asymptotically similar, according to definition 3, to a limit problem:

$$(S_L) \qquad \begin{cases} \frac{dZ_L}{ds} = G_L(Z_L), & 0 < s \leq s_L, \\ Z_L(0) = 0, & \\ H\left(Z_L(s)\right) \neq 0, \forall s < s_L \text{ and} & H\left[Z_L(s_L)\right] = 0, \end{cases}$$

2. *the sequence* $\{s_n\}$ *of resulting end-of-slice rescaled times is uniformly upper bounded:*

$$\forall n \geq 1, \ \exists \bar{s} > 0, \quad s_n \leq \bar{s}. \tag{1.29}$$

3. *there exists* $\hat{S} > 0$ *such that:*

$$\begin{cases} \forall s \in [0, s_L], & Z_L(s) \in B_{\hat{S}}, \\ \forall n, \ \ \forall s \in [0, s_n], & Z_n(s) \in B_{\hat{S}}, \end{cases} \tag{1.30}$$

where the domain $B_{\hat{S}}$ *is defined by:* $B_{\hat{S}} = \left\{W \in \mathbb{R}^K, \ \|W\|_\infty \leq \hat{S}\right\}$,

4. G_L *verifies a Lipschitz condition (with constant* κ*) on the domain* $B_{\hat{S}}$*, i.e.:*

$$\forall W_1, W_2 \in B_{\hat{S}}, \quad ||G_L(W_1) - G_L(W_2)||_\infty \leq \kappa ||W_1 - W_2||_\infty,$$

then, the sequence of functions $\{Z_n(.)\}$ *that solves the rescaled systems* (S'_n) *converges uniformly to the solution* $Z_L(.)$ *of the limit system* (S_L)*, on* $[0, \hat{s}]$*:*

$$\lim_{n \to \infty} \left[\max_{s \in [0, \hat{s}]} ||Z_n(s) - Z_L(s)||_\infty \right] = 0, \tag{1.31}$$

where:

$$\hat{s} = \max\left(s_L, \max_n s_n \leq \bar{s}\right). \tag{1.32}$$

Proof:
First, the solution Z_L of the limit problem and the solutions Z_n of the rescaled systems are extended beyond their EOS condition, for them all to be defined onto the maximal interval $[0, \hat{s}]$.
For all n and for all $s \in [0, \hat{s}]$, one has:
$Z_n(s) - Z_L(s) = \int_0^s \left[G_n\left(Z_n(\tau)\right) - G_L\left(Z_L(\tau)\right)\right] d\tau$, yielding:
$||Z_n(s) - Z_L(s)||_\infty \leq s \max_{[0,s]} \left\|G_n\left(Z_n(s)\right) - G_L\left(Z_L(s)\right)\right\|_\infty$. But:
$\left\|G_n\left(Z_n(s)\right) - G_L\left(Z_L(s)\right)\right\|_\infty = \left\|G_n\left(Z_n(s)\right) - G_L\left(Z_n(s)\right) + G_L\left(Z_n(s)\right) - G_L\left(Z_L(s)\right)\right\|_\infty$,
implying:
$\left\|G_n\left(Z_n(s)\right) - G_L\left(Z_L(s)\right)\right\|_\infty \leq \left\|G_n\left(Z_n(s)\right) - G_L\left(Z_n(s)\right)\right\|_\infty + \left\|G_L\left(Z_n(s)\right) - G_L\left(Z_L(s)\right)\right\|_\infty$.

Using now the Lipschitz property of G_L:
$\|G_n(Z_n(s)) - G_L(Z_L(s))\|_\infty \leq \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty + \kappa \|Z_n(s) - Z_L(s)\|_\infty$, yields for all n and for all $s \in [0, \hat{s}]$:

$$\begin{array}{r} ||Z_n(s) - Z_L(s)||_\infty \leq s \max_{[0,s]} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty \\ + s\kappa \max_{[0,s]} \|Z_n(s) - Z_L(s)\|_\infty . \end{array} \tag{1.33}$$

Let now $s_1 = \frac{1}{2\kappa}$ (implying $s_1\kappa = \frac{1}{2} < 1$).
Case 1: $s_1 \geq \hat{s}$ (yielding $\hat{s}\kappa < 1$)
Since the previous inequality holds for all $s \in [0, \hat{s}]$, it follows for all n and by making $s = \hat{s}$ in (1.33):
$\max_{[0,\hat{s}]} ||Z_n(s) - Z_L(s)||_\infty \leq \hat{s} \max_{[0,\hat{s}]} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty$

$$+\hat{s}\kappa \max_{[0,\hat{s}]} \|Z_n(s) - Z_L(s)\|_\infty,$$

yielding:
$(1 - \hat{s}\kappa) \max_{[0,\hat{s}]} \|Z_n(s) - Z_L(s)\|_\infty \leq \hat{s} \max_{[0,\hat{s}]} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty$.
Since $1 - \hat{s}\kappa > 0$, one deduces for all n:
$\max_{s\in[0,\hat{s}]} \|Z_n(s) - Z_L(s)\|_\infty \leq \frac{\hat{s}}{(1-\hat{s}\kappa)} \max_{[0,\hat{s}]} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty$.
Taking now into consideration assumption (1.30) yields:
$\max_{s\in[0,\hat{s}]} \|Z_n(s) - Z_L(s)\|_\infty \leq \frac{\hat{s}}{(1-\hat{s}\kappa)} \max_{Z_n(s)\in B_{\hat{S}}} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty$
and therefore, using the uniform convergence of G_n toward G_L, on $B_{\hat{S}}$:
$\lim_{n\to\infty} \left[\max_{s\in[0,\hat{s}]} ||Z_n(s) - Z_L(s)||_\infty\right] = 0$.
Hence the uniform convergence of $\{Z_n\}$ toward Z_L.
Case 2: $s_1 \leq \hat{s}$
Applying inequality (1.33) onto s_1 yields, in the same way and for all n:
$\max_{[0,s_1]} \|Z_n(s) - Z_L(s)\|_\infty \leq \frac{s_1}{(1-s_1\kappa)} \max_{[0,s_1]} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty$.
making Z_n uniformly convergent toward Z_L on $[0, s_1]$.
Then, repeating recurrently the previous procedure, for all n and for all $s \in [s_1, \hat{s}]$ gives:
$Z_n(s) - Z_L(s) = \int_{s_1}^{s} [G_n(Z_n(\tau)) - G_L(Z_L(\tau))]\, d\tau$, yielding:
$||Z_n(s) - Z_L(s)||_\infty \leq (s - s_1) \max_{[s_1,s]} \|G_n(Z_n(s)) - G_L(Z_L(s))\|_\infty$, and therefore:

$$\begin{array}{r} ||Z_n(s) - Z_L(s)||_\infty \leq (s - s_1) \max_{[s_1,s]} \|G_n(Z_n(s)) - G_L(Z_n(s))\|_\infty \\ + (s - s_1)\kappa \max_{[s_1,s]} \|Z_n(s) - Z_L(s)\|_\infty . \end{array}$$

Letting now $s_2 = s_1 + \frac{1}{2\kappa}$ (implying $(s_2 - s_1)\kappa = \frac{1}{2} < 1$), 2 cases should be considered:
If $s_2 \geq \hat{s}$ (yielding $(\hat{s} - s_1)\kappa < 1$), then the uniform convergence of Z_n toward Z_L on $[s_1, s_2]$ (and therefore on $[s_1, \hat{s}]$) is deduced in the same way.
If $s_2 < \hat{s}$, repeat again and let $s_3 = s_2 + \frac{1}{2\kappa}$... until an interval $[s_{i-1}, s_i]$ of uniform convergence, such that $s_i \geq \hat{s}$, is reached. ■

Corollary 1 *: Convergence of* $\mathbf{s_n}$ *and* $\mathbf{Z_n(s_n)}$
If each of the functions $H[Z_L(.)]$ *and* $\{H[Z_n(.)]\}$ *is assumed to change sign exactly once on* $[0, \hat{s}]$*, making* 0 *having exactly one preimage in* $[0, \hat{s}]$*, namely* s_L *and* $\{s_n\}$ *respectively, and if there exists* $\epsilon > 0$ *such that there exist bijections mapping an interval containing* s_L *and, for every* n*, an interval containing* s_n*, onto the interval* $[-\epsilon, \epsilon] \subset \mathbb{R}$*,*
then, under the assumptions of theorem 1, one has:

$$\left\{ \begin{array}{l} lim_{n\to\infty} s_n = s_L, \\ lim_{n\to\infty} Z_n(s_n) = Z_L(s_L). \end{array} \right. \tag{1.34}$$

Proof:

- Theorem 1 proves the uniform convergence of Z_n toward Z_L, on $[0, \hat{s}]$, implying: $\forall s \in [0, \hat{s}], \quad \lim_{n\to\infty} Z_n(s) = Z_L(s)$.
  It follows, by continuity of the function H, the convergence of the functions $\{H \circ Z_n\}$ toward $H \circ Z_L$: $\forall s \in [0, \hat{s}], \quad \lim_{n\to\infty} H\left[Z_n(s)\right] = H\left[Z_L(s)\right]$.
  Consider now the restrictions of the functions $\{H \circ Z_n\}$ and $H \circ Z_L$ onto the intervals containing $\{s_n\}$ and s_L respectively, where they define bijections toward $[-\epsilon, \epsilon] \subset \mathbb{R}$, as assumed.
  The convergence of the bijections $\{H \circ Z_n\}$ toward $H \circ Z_L$ yields the convergence of their inverse bijections defined on $[-\epsilon, \epsilon]$.
  Since $0 \in [-\epsilon, \epsilon]$, it follows that:
  $\lim_{n\to\infty} \left[H \circ Z_n\right]^{-1}(0) = \left[H \circ Z_L\right]^{-1}(0)$, or equivalently: $\lim_{n\to\infty} s_n = s_L$.

- One has $\left\|Z_n(s_n) - Z_L(s_L)\right\|_\infty \leq \left\|Z_n(s_n) - Z_n(s_L)\right\|_\infty + \left\|Z_n(s_L) - Z_L(s_L)\right\|_\infty$.
  But $Z_n(s_n) - Z_n(s_L) = \int_{s_L}^{s_n} G_n\left[Z_n(s)\right] ds$, yields by uniform similarity:
  $\left\|Z_n(s_n) - Z_n(s_L)\right\|_\infty \leq C\left|s_n - s_L\right|$,
  and $\lim_{n\to\infty} s_n = s_L$ implies then: $\lim_{n\to\infty} \left\|Z_n(s_n) - Z_n(s_L)\right\|_\infty = 0$.
  Besides, the uniform convergence of Z_n toward Z_L gives:
  $\lim_{n\to\infty} \left\|Z_n(s_L) - Z_L(s_L)\right\|_\infty = 0$. It follows that:
  $\lim_{n\to\infty} \left\|Z_n(s_n) - Z_L(s_L)\right\|_\infty = 0$, or equivalently: $\lim_{n\to\infty} Z_n(s_n) = Z_L(s_L)$.

■

1.3.3 Weak Similarity

Definition 4 ***: Weak Similarity***
The rescaled systems (1.18):

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n \\ Z_n(0) = 0, & \\ H\left(Z_n(s)\right) \neq 0, \forall s < s_n \ \ and & H\left[Z_n(s_n)\right] = 0, \end{cases}$$

are said to present a weak similarity on n_r consecutive slices, starting at slice n_0, if the functions $\{H_n\}$ governing the rescaled EOS condition are invariant and if:

$$\exists \epsilon << 1, \ \ \forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \quad \left\|Z_n(s_n) - Z_{n-1}(s_{n-1})\right\|_\infty < \epsilon. \tag{1.35}$$

We have numerically noticed that a relevant choice of an EOS condition that suits the global behavior of the physical problem, together with a weak similarity of the rescaled systems on n_r consecutive slices, starting at slice n_0, make this weak similarity hold on a number of slices that is much larger than n_r.

This case of weak similarity is encountered, for example, when dealing with unsolvable mathematical problems deriving from celestial mechanics, like those describing the motion of a satellite. When the gravitational attraction of the earth is the only force applied to the satellite, the resulting trajectory is a simple Keplerian ellipse. But many other forces perturb the motion. However, they are comparatively small and are known to result in slight variations, changing the Keplerian ellipse into an osculating ellipse, i.e. an ellipse of which the parameters are no more constant and are slightly different at each instant (see chapter 7).

It is clear that invariance and asymptotic similarity are particular (and stronger) cases of weak similarity.

1.3.4 Uniform Similarity

Definition 5 ***: Uniform Similarity of the Initial Value Problems***
Let $J_{G_n}(.) : \mathbb{R}^K \to \mathbb{R}^K \times \mathbb{R}^K$ be the Jacobian of G_n. The rescaled initial value problems:

$$(IVP_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

are said to be uniformly similar, if the selection of the time-rescaling parameter β_n leads to:

$$\forall n \geq 1,\ \forall S > 0,\ \exists C, C' > 0, \quad ||W||_\infty \leq S \implies \begin{cases} ||G_n(W)||_\infty \leq C, \\ ||J_{G_n}(W)||_\infty \leq C'. \end{cases} \tag{1.36}$$

It should be noted that C and C' depend on S but are independent of n.

Definition 6 ***: Uniform Similarity of*** (S'_n)
The rescaled initial value shooting problems:

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, & \\ H\left[Z_n(s)\right] \neq 0,\ \forall s < s_n \ \text{ and } \ H\left[Z_n(s_n)\right] = 0, & \end{cases}$$

are said to be uniformly similar, if the choice of the EOS condition governed by the function H, together with the selection of the time-rescaling parameters $\{\beta_n\}$, are such that:

1. *the initial value problems deriving from (S'_n):*

$$(IVP_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

   are uniformly similar, according to definition 5,

2. *the sequence $\{s_n\}$ of resulting end-of-slice rescaled times is uniformly bounded (upper and lower), independently of n:*

$$\exists \underline{s},\ \bar{s} > 0,\ \forall n \geq 1, \quad \underline{s} \leq s_n \leq \bar{s}. \tag{1.37}$$

One should note that the property of uniform similarity results in "similar numerical simulations" (i.e. same order of magnitude of the computations) on all time slices.
Besides, it favors the local approach of a numerical method in solving the rescaled systems, whereas one can control the growth of $\|G_n\|_\infty$ and $\|J_{G_n}\|_\infty$, i.e. control the stiffness of the problem. Such local property follows directly from the definition of this type of similarity which infers the uniform boundedness of $\|G_n(W)\|_\infty$ and $\|J_{G_n}(W)\|_\infty$, when $\|W\|_\infty < S$. Hence, by introducing an End-Of-Slice condition that could insure the uniform boundedness of $\|Z_n\|_\infty$, one gets directly the uniform boundedness of $\|G_n\|_\infty$ and $\|J_{G_n}\|_\infty$.

The lower and upper boundedness are discussed in Chapter 2 in the case of explosive problems.
The following theorem intends to provide a critical choice of $\{\beta_n\}$that yields the uniform similarity of (IVP_n).

Theorem 2 ***: Critical choice of*** $\{\beta_n\}$
If for any $S > 0$*, we let for all* $n \geq 1$*:*

$$C_{n,1} = \max_{\|Z_n\|<S} \left\| D_n^{-1} F\big(Y_{n-1} + D_n Z_n\big) \right\|_\infty , \tag{1.38}$$

$$C_{n,2} = \max_{\|Z_n\|<S} \left\| D_n^{-1} \left[J_F(Y_{n-1} + D_n Z_n) \right] Y_{n-1} \right\|_\infty , \tag{1.39}$$

then, the critical choice:

$$\forall n \geq 1, \;\; \beta_n = \frac{1}{\max\{C_{n,1}, C_{n,2}\}} \tag{1.40}$$

yields the uniform similarity of the initial value problems:

$$(IVP_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

Proof.
Knowing that:

$$\forall i \in \{1, 2, \cdots, K\}, \quad \left(G_n(Z_n)\right)_i = \beta_n D_n^{-1} F_i\big(Y_{n-1} + D_n Z_n\big),$$

one deduces:

$$\forall i, m \in \{1, 2, \cdots, K\}, \quad \left[J_{G_n}(Z_n)\right]_{im} = \frac{\partial \left(G_n(Z_n)\right)_i}{\partial Z_{n,m}} = \beta_n D_n^{-1} \frac{\partial \left(F(Y)\right)_i}{\partial Y_m} \frac{\partial Y_m}{\partial Z_{n,m}},$$

with $\frac{\partial Y_m}{\partial Z_{n,m}} = \begin{cases} Y_{n-1,m} & \text{if } Y_{n-1,m} \neq 0, \\ 1 & \text{if } Y_{n-1,m} = 0. \end{cases}$
Let us assume, without loss of generality, that the nonzeroness condition (1.11) is satisfied. It follows :

$$\left[J_{G_n}(Z_n)\right]_{im} = \beta_n D_n^{-1} \left[J_F(Y)\right]_{im} Y_{n-1,m},$$

and therefore:

$$\left\| J_{G_n}(Z_n) \right\|_\infty = \beta_n \left\| D_n^{-1} \left[J_F(Y_{n-1} + D_n Z_n) \right] Y_{n-1} \right\|_\infty ,$$

with:

$$\left\| G_n(Z_n) \right\|_\infty = \beta_n \left\| D_n^{-1} F\big(Y_{n-1} + D_n Z_n\big) \right\|_\infty .$$

Clearly, the choice (1.40) for $\{\beta_n\}$ yields:

$$\forall n \geq 1, \;\; \forall S > 0, \quad ||Z_n||_\infty \leq S \implies \begin{cases} ||G_n(Z_n)||_\infty \leq 1, \\ ||J_{G_n}(Z_n)||_\infty \leq 1. \end{cases}$$

Hence, the resulting uniform similarity of (IVP_n).

■

This way for finding a "critical" value for the time-rescaling parameter, yielding a uniform similarity of (IVP_n), is explicitly described in chapter 3, in the case where F has the algebraic form, component wise given by: $(F(Y))_i = \sum_j a_{ij} (Y_i)^{k_{ij}} (Y_j)^{l_{ij}}$.

Chapter 2

Selection of an EOS Condition

Initially, our time-sliced computation has been motivated by the need to make the general problem (S) equivalent to a sequence of local problems (S_n) on which robust methods can be devised, particularly for solving stiff systems that could be explosive. In line with this, rescaling problems (S_n) into (S'_n) in a way to get uniform similarity, has proved to yield robust algorithms.

In this thesis, we intend to use the same time-sliced computation in a way to make problem (S) equivalent to a sequence of local problems (S_n) that could be solved in parallel. As it will be shown in Chapter 4, rescaling problems (S_n) into (S'_n) in a way to get invariance or near-invariance (i.e. asymptotic or even weak similarity), could provide a prediction procedure for the starting values of the solution on each slice thus, allowing to solve (S'_n) in a parallel-in-time way.

In general, and under assumption 1, given in Chapter 1, one can find more than one possible EOS condition for a problem which solution has a given global behavior. The one providing the desired type of similarity is then selected.

At this point of the development of our method, we have defined and experimented EOS conditions for problems which solution has one of the two following behaviors: (i) explosive behavior and (ii) oscillatory behavior.
We know also how to define EOS conditions for other cases such as problems having an extinctive solution or which solution has an attraction point at infinity (that could be zero, reducing the problem to an extinctive one). However, we did not experiment our method on such problems, yet.

2.1 Case of Explosive Problems

In case of explosive problems:

$$\lim_{t\to\infty} \|Y(t)\|_\infty = \infty,$$

the choices of EOS condition proposed in this section aim at limiting the growth of the solution.

2.1.1 General Choice

A possible choice of the functions $\{E_n\}$, governing the EOS condition, is obtained by defining E_n, for a given $S > 0$ and for all n, in terms of the starting value Y_{n-1}, as follows:

$$\forall W \in \mathbb{R}^K, \quad E_n(W) = \|W - Y_{n-1}\|_\infty - S\,\|Y_{n-1}\|_\infty . \tag{2.1}$$

For all $S > 0$, the EOS condition (1.2):

$$\begin{array}{ll} \text{at } t = T_n, & E_n\left[Y\left(T_n\right)\right] = 0, \\ \forall t \in \left(T_{n-1}, T_n\right), & E_n\left[Y\left(t\right)\right] \neq 0, \end{array}$$

bounds the solution $Y(t)$, on each n^{th} slice, within the closed hyperball of $\mathbb{R}^K$, centered at Y_{n-1} and of radius $\rho_n = S\,\|Y_{n-1}\|_\infty$ using the infinity norm:

$$B_{\rho_n}[Y_{n-1}] = \left\{Y \in \mathbb{R}^K,\ \|Y - Y_{n-1}\|_\infty \leq \rho_n\right\}. \tag{2.2}$$

For a given $S > 0$, the n^{th} slice is then ended as soon as the solution satisfies:

$$\begin{array}{c} \text{at } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \|Y\left(T_n\right) - Y_{n-1}\|_\infty = S\,\|Y_{n-1}\|_\infty , \\ \text{with} \quad \|(Y(t) - Y_{n-1})\|_\infty < S\,\|Y_{n-1}\|_\infty , \quad \text{if } T_{n-1} < t < T_n. \end{array} \tag{2.3}$$

Note that, for all $S > 0$ and because of the continuity of the solution $Y(.)$ and its explosive behavior, one has $E_n\left[Y(t)\right] = \|Y(t) - Y_{n-1}\|_\infty - S\,\|Y_{n-1}\|_\infty$ continuously varying from $E_n\left[Y(T_{n-1})\right] = -S\,\|Y_{n-1}\|_\infty$ to $\lim_{t\to\infty} E_n\left[Y(t)\right] = \infty$.
This makes the set $\Delta_n = \{t > T_{n-1} \,|\, E_n\left[Y(t)\right] = 0\}$ not empty and therefore, the EOS condition (2.3) is guaranteed to be reached at $T_n = \min\{\Delta_n\}$.

Rescaled EOS condition:
Applied to $Y(t)$ $(t > T_{n-1})$, the function E_n defined in (2.1) yields: $E_n\left[Y\left(t\right)\right] = \|Y(t) - Y_{n-1}\|_\infty - S\,\|Y_{n-1}\|_\infty$ and is rescaled as:

$$E_n\left[Y_{n-1} + D_n Z_n(s)\right] = \|D_n Z_n(s)\|_\infty - S\,\|Y_{n-1}\|_\infty .$$

This yields a family $\{H_n\}$ of functions, governing the rescaled EOS condition, given by:

$$\forall W \in \mathbb{R}^K,\ H_n(W) = \|D_n W\|_\infty - S\,\|Y_{n-1}\|_\infty . \tag{2.4}$$

Clearly, the functions $\{H_n\}$ are parametrized by the corresponding starting values $\{Y_{n-1}\}$ of each time-slice and therefore, they are not invariant according to definition 1.
For a given cutoff value $S > 0$, the n^{th} slice is ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{array}{c} \text{at } s = s_n, \quad H_n\left[Z_n(s_n)\right] = 0 \iff \|D_n Z_n(s_n)\|_\infty = S\,\|Y_{n-1}\|_\infty , \\ \text{with} \quad \|D_n Z_n(s)\|_\infty < S\,\|Y_{n-1}\|_\infty , \quad \text{if } 0 < s < s_n. \end{array} \tag{2.5}$$

2.1.2 Case of nonzeroness of $\{Y_n\}$

In case the nonzeroness condition (1.11) is satisfied:

$$\forall n, \forall i : 1 \leq i \leq k,\ Y_{n,i} \neq 0,$$

another choice would be to define E_n for all $W \in \mathbb{R}^K$, in terms of the starting value Y_{n-1}, by:

$$\forall W \in \mathbb{R}^K, \qquad E_n(W) = \max_{i=1,2,\ldots,K} \left| \frac{W_i - Y_{n-1,i}}{Y_{n-1,i}} \right| - S.$$

By letting, on each n^{th} slice, D_n be the diagonal matrix having Y_{n-1} on its main diagonal, the expression of E_n can equivalently be written as:

$$E_n(W) = \left\| D_n^{-1} \left(W - Y_{n-1} \right) \right\|_\infty - S. \tag{2.6}$$

For all $S > 0$, the EOS condition (1.2) acts on the absolute value of the relative variation of each component of $Y(t)$ relatively to the same component of Y_{n-1} and prevents it from exceeding a cut-off value S.
For a given $S > 0$, the n^{th} slice is then ended as soon as the solution satisfies:

$$\begin{array}{c} \text{at } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \left\| D_n^{-1} \left(Y(T_n) - Y_{n-1} \right) \right\|_\infty = S, \\ \text{with} \quad \left\| D_n^{-1} \left(Y(t) - Y_{n-1} \right) \right\|_\infty < S, \quad \text{if } \ T_{n-1} < t < T_n. \end{array} \tag{2.7}$$

Here also, for all $S > 0$ and because of the continuity of the solution $Y(.)$ and its explosive behavior, one has $E_n\left[Y(t)\right] = \left\| D_n^{-1} \left(Y(t) - Y_{n-1} \right) \right\|_\infty - S$ continuously varying from $E_n\left[Y(T_{n-1})\right] = -S$ to $\lim_{t \to \infty} E_n\left[Y(t)\right] = \infty$.
This makes the set $\Delta_n = \{t > T_{n-1} \,|\, E_n\left[Y(t)\right] = 0\}$ not empty and therefore, the EOS condition (2.7) is guaranteed to be reached at $T_n = \min\{\Delta_n\}$.

Particular Scalar Case:
A particular case would be the scalar case, yielding: $\forall n,\ \left| \frac{Y_n - Y_{n-1}}{Y_{n-1}} \right| = S$, that translates, in case the solution Y has a constant sign with $|Y|$ monotonously increasing, to:

$$\forall n, \quad \frac{Y_n}{Y_{n-1}} - 1 = S \qquad \Longleftrightarrow \qquad \forall n, \quad \frac{Y_n}{Y_{n-1}} = 1 + S = \text{ constant},$$

making $\{Y_n\}$ a geometric sequence, with constant ratio equal to $(1 + S) > 1$.

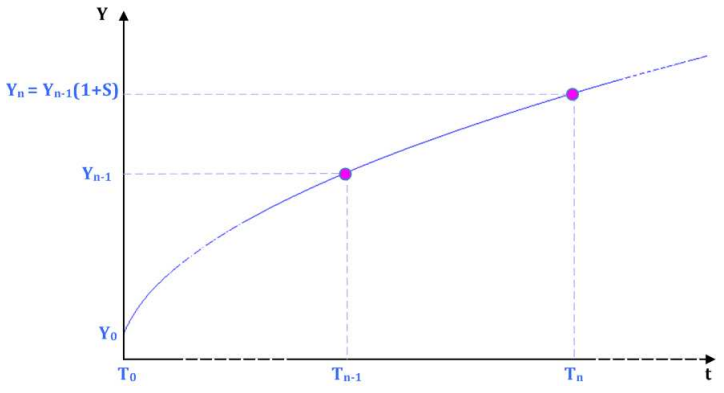


Figure 2.1: EOS Condition: Scalar Explosive Positive Increasing Problem

Rescaled EOS condition:
Under the assumed nonzeroness condition (1.11), the change of variable is given in (1.12):

$$Y(t) = D_n \left(\mathbf{1} + Z_n(s)\right)$$

where D_n is the diagonal matrix having Y_{n-1} on its main diagonal and $\mathbf{1}$ is the vector of ones in $\mathbb{R}^K$.
Applied to $Y(t)$ $(t > T_{n-1})$, the function E_n defined in (2.6) yields: $E_n\left[Y\left(t\right)\right] = \left\|D_n^{-1}\left(Y(t) - Y_{n-1}\right)\right\|_\infty - S$ and simplifies after rescaling to:

$$E_n\left[D_n\left(\mathbf{1} + Z_n(s)\right)\right] = \left\|Z_n(s)\right\|_\infty - S.$$

Hence, the Invariance of $\{H_n\}$, according to definition 1, since a unique function $H : \mathbb{R}^K \to \mathbb{R}$ governs the rescaled EOS condition and is defined, independently of n, by:

$$\forall W \in \mathbb{R}^K,\ H(W) = \left\|W\right\|_\infty - S. \tag{2.8}$$

For a given cutoff value $S > 0$, the n^{th} slice is then ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{array}{ccccc} \text{at } s = s_n, & & H\left[Z_n(s_n)\right] = 0 & \Longleftrightarrow & \left\|Z_n(s_n)\right\|_\infty = S, \\ & \text{with} & \left\|Z_n(s)\right\|_\infty < S, & \text{if } 0 < s < s_n. & \end{array} \tag{2.9}$$

Remark 2 *If, moreover, $||Y(t)||_\infty$ monotonously increases with t, one has:*
$\forall t > T_n,\ \left\|D_n^{-1}\left(Y(t) - Y_{n-1}\right)\right\|_\infty - S > 0$ *and therefore,* $\forall s > s_n,\ \left\|Z_n(s)\right\|_\infty > S$.
This makes the additional assumption of the existence of a bijection, required in corollary 1 of theorem 1, automatically satisfied when the time-slicing is done using EOS condition (2.9), (2.28), since it implies:

$$\begin{array}{ll} \textit{at } s = s_n, & \left\|Z_n(s_n)\right\|_\infty = S, \\ \textit{if } 0 < s < s_n, & \left\|Z_n(s_n)\right\|_\infty < S, \\ \forall s > s_n, & \left\|Z_n(s_n)\right\|_\infty > S, \end{array}$$

making the function $H\left[Z_n(.)\right]$ define a bijection from $[0, \infty)$ toward $[-S, \infty)$.

Proposition 1 ***: Lower boundedness of*** $\{s_n\}$
If the selection of $\{\beta_n\}$ is such that:

$$\exists C,\ \forall n \geq 1, \qquad \max_{\left\|Z\right\|_\infty \leq S} \left\|G_n(Z)\right\|_\infty \leq C, \tag{2.10}$$

then, the EOS condition (2.9) yields a sequence $\{s_n\}$ of rescaled time values at the end of slices that is uniformly lower bounded, i.e.:

$$\exists \underline{s} > 0,\ \forall n \geq 1, \qquad \underline{s} \leq s_n. \tag{2.11}$$

Proof.
From $\frac{dZ_n}{ds} = G_n\left[Z_n(s)\right]$, one deduces:

$$\forall n,\ \forall s \in [0, s_n] \qquad Z_n(s) = \int_0^s G_n\left[Z_n(\sigma)\right] d\sigma,$$

yielding:

$$\forall n,\ \forall s \in [0, s_n] \qquad \|Z_n(s)\|_\infty \leq \int_0^s \|G_n\left[Z_n(\sigma)\right]\|_\infty d\sigma \leq Cs \leq Cs_n.$$

At $s = s_n$, the EOS condition (2.9) leads to $\|Z_n(s_n)\|_\infty = S$ and therefore:

$$\forall n, \qquad S \leq Cs_n \qquad \Longleftrightarrow \qquad \underline{s} = \frac{S}{C} \leq s_n.$$

■

Theorem 3 ***Upper boundedness of*** $\{s_n\}$
If, in case of explosive problems, one has:

1. $||Y(t)||_\infty$ *monotonously increases with* t *and* $\forall i,\ \forall t,\ Y_i(t) > 0$,
2. $\exists i_0 \in \{1, 2, \cdots, K\}$, $\begin{cases} \forall i,\ \forall t, \quad Y_i(t) \leq Y_{i_0}(t), \\ \lim_{t\to\infty} Y_{i_0}(t) = \infty, \\ \lim_{t\to\infty} F_{i_0}\left(Y(t)\right) = \infty, \\ F_{i_0}\left(Y(t)\right) \text{ and } Y_{i_0}^{-1} F_{i_0}\left(Y(t)\right) \text{ are positive and increasing,} \end{cases}$
3. *For a given* $c_0 > 0$, *the sequence* $\{\beta_n\}$ *is selected such as:*

$$\forall n \geq 1, \quad \beta_n = \frac{c_0 Y_{n-1,i_0}}{F_{i_0}\left(Y_{n-1}\right)} = \frac{c_0}{\left\|D_n^{-1} F(Y_{n-1})\right\|_\infty}, \tag{2.12}$$

then, the sequence $\{s_n\}$ *of rescaled times, resulting from the EOS condition (2.7), is uniformly upper bounded (independently of* n*):*

$$\forall n \geq 1,\ \exists \bar{s} > 0, \quad s_n \leq \bar{s}. \tag{2.13}$$

Proof.
For any EOS condition generating a coarse grid $\{T_n\}$ starting at $T_0 = 0$, the explosive behavior of the solution:

$$\lim_{t\to\infty} \|Y(t)\|_\infty = \infty,$$

together with assumption 1 of the theorem, yields:

$$0 < ||Y_0||_\infty \leq ||Y_1||_\infty \leq ||Y_2||_\infty \leq \cdots \leq ||Y_n||_\infty \leq \cdots \tag{2.14}$$

Besides, since $Z_n(s) = D_n^{-1} Y(t) - \mathbf{1}$, it follows from the same assumption 1 of the theorem, that $Z_{n,i}(s) > 0$ $(\forall i)$, $\lim_{s\to\infty} \|Z_n(s)\|_\infty = \infty$ and $||Z_n(s)||_\infty$ monotonously increases with s. Combined with assumption 2 of the theorem, those results make:

$$G_{i_0}\left[Z_n(s)\right] = \beta_n Y_{n-1,i_0}^{-1} F_{i_0}\left(D_n\left(\mathbf{1} + Z_n(s)\right)\right)$$

monotonously increasing with s. It follows: $\forall n,\ \forall s > 0\ \ G_{n,i_0}\left[Z_n(s)\right] \geq G_{n,i_0}\left[Z_n(0)\right]$, i.e.:

$$\forall n,\ \forall s > 0 \qquad G_{n,i_0}\left[Z_n(s)\right] \geq \beta_n Y_{n-1,i_0}^{-1} F_{i_0}\left(Y_{n-1}\right).$$

The choice (2.12) for $\{\beta_n\}$ yields then, for a given $c_0 > 0$:

$$\forall n \geq 1,\ \forall s > 0 \quad G_{n,i_0}\left[Z_n(s)\right] \geq c_0 > 0, \tag{2.15}$$

Besides, and because $\frac{dZ_{n,i_0}}{ds} = G_{n,i_0}\left[Z_n(s)\right]$, one deduces:

$$\forall n,\ \ \forall s > 0 \qquad Z_{n,i_0}(s) = \int_0^s G_{n,i_0}\left[Z_n(\sigma)\right] d\sigma \geq c_0 s.$$

Applying this to $s = s_n$, where $\forall n,\ Z_{n,i_0}(s_n) = S$ (since the EOS condition is reached at the component $i_0,\ \forall n$, due to assumption 2 of the theorem), one concludes:

$$\forall n, \qquad s_n \leq \frac{S}{c_0} = \bar{s}.$$

■

The following is a corollary to both proposition 1 and theorem 3.

Corollary 2 *If, under the assumptions of theorem 3, one has:*

$$\exists c_0 > 0, \quad \forall n, \qquad \frac{c_0}{\left\|D_n^{-1} F(Y_{n-1})\right\|_\infty} \leq \frac{1}{\max\left\{C_{n,1}, C_{n,2}\right\}}, \tag{2.16}$$

where $C_{n,1}$ and $C_{n,2}$ are given in formulae (1.38) and (1.39) of theorem 2 in Chapter 1, then the choice (2.12) for $\{\beta_n\}$, together with the EOS condition (2.7), result, for a given S, in rescaled problems:

$$(S'_n) \quad \left\{ \begin{array}{ll} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, & \\ \left\|Z_n(s)\right\|_\infty < S,\ \forall s < s_n \ \ and \ \ \left\|Z_n(s_n)\right\|_\infty = S, & \end{array} \right.$$

that are uniformly similar, according to definition 6 in Chapter 1.

The existence of such c_0 is proved in theorem 5 of Chapter 3, in the particular case of a linear problem.

Proof.

- If there exists c_0 satisfying (2.16), then the choice (2.12) for $\{\beta_n\}$ yields:

$$\forall n \geq 1, \qquad \beta_n = \frac{c_0}{\left\|D_n^{-1} F(Y_{n-1})\right\|_\infty} \leq \frac{1}{\max\left\{C_{n,1}, C_{n,2}\right\}}.$$

  This makes the proof of theorem 2 of Chapter 1 keep holding, yielding the uniform similarity of the initial value problems (IVP_n) $\left\{ \begin{array}{l} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0. \end{array} \right.$

- Together with the EOS condition (2.7), this uniform similarity of (IVP_n) implies the lower uniform boundedness of $\{s_n\}$ (by proposition 1).

- The upper uniform boundedness of $\{s_n\}$ is directly deduced from theorem 3.

Hence, the uniform similarity of the rescaled problems (S'_n), according to definition 6 in Chapter 1. ■

2.2 Case of Oscillatory solutions

When the behavior of the solution is oscillatory, over a long period of time, in the sense that there exists a two-dimensional plane P in $\mathbb{R}^K$ in which the projection of the solution rotates about a fixed center ω, a slice is ended when the solution completes a full, or almost full, rotation in that plane. Let P be the $i_1 i_2$-plane and assume, without loss of generality, that this rotation occurs about the origin ω.

2.2.1 General Choice

1. **Definition of a Rotation Angle $\theta_n[Y(t)]$:**
   Let $M(t) = (Y_{i_1}(t), Y_{i_2}(t))$ be the projection of the solution $Y(t)$ on the plane P, at each instant t. The projection of the starting value Y_{n-1} is then a starting point: $M_{n-1} = (Y_{i_1}(T_{n-1}), Y_{i_2}(T_{n-1})) = (Y_{n-1,i_1}, Y_{n-1,i_2})$.
   While rotating about ω, the point $M(t)$ defines, in the plane P, a monotonously increasing rotation angle:

$$\theta_n[Y(t)] = \left|\left(\omega\vec{M}_{n-1}, \omega\vec{M}(t)\right)\right|. \tag{2.17}$$

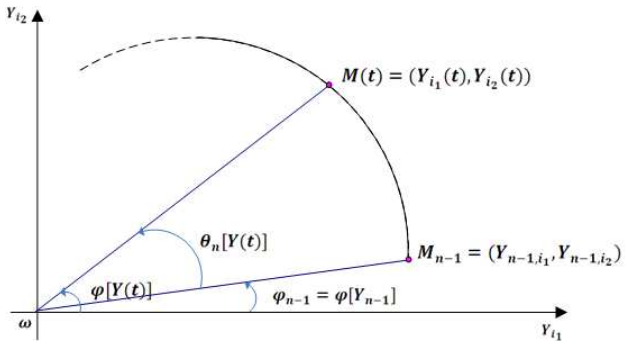


Figure 2.2: Oscillating Problem: Rotation Angle

2. **Evaluation of $\theta_n[Y(t)]$:**
   A function ψ_n, intended to apply on $Y(t)$ through its i_1, i_2-components for getting $\cos(\theta_n[Y(t)])$, is defined in terms of the dot product and the euclidean norm in $\mathbb{R}^2$ as:

$$\psi_n[Y(t)] = \cos\left(\omega\vec{M}_{n-1}, \omega\vec{M}(t)\right) = \frac{\omega\vec{M}_{n-1} \cdot \omega\vec{M}(t)}{\left\|\omega\vec{M}_{n-1}\right\|_2 \left\|\omega\vec{M}(t)\right\|_2}, \tag{2.18}$$

   or more explicitly:

$$\psi_n[Y(t)] = \frac{Y_{n-1,i_1}Y_{i_1}(t) + Y_{n-1,i_2}Y_{i_2}(t)}{\sqrt{Y_{n-1,i_1}^2 + Y_{n-1,i_2}^2}\sqrt{Y_{i_1}^2(t) + Y_{i_2}^2(t)}}.$$

   Since the function "$\cos^{-1}$" gives values ranging in $[0, \pi]$, then a way for defining a continuously increasing rotation angle $\theta_n[Y(t)]$ in the present case where the projection of the solution keeps rotating about the origin ω, is done using the following procedure on successive intervals of range π.

Definition 7 *: Procedure for getting* $\theta_{\mathbf{n}}[\mathbf{Y(t)}] = \left|\left(\omega\tilde{\mathbf{M}}_{\mathbf{n-1}}, \omega\tilde{\mathbf{M}}(\mathbf{t})\right)\right|$

(a) At $t = T_{n-1}$*:* $\theta_n[Y(t)] = 0$.

(b) When $T_{n-1} < t < T_{n-\frac{1}{2}}$*,* $\theta_n[Y(t)]$ *increases from* 0 *to* π *and is given by:*

$$\theta_n[Y(t)] = \cos^{-1}(\psi_n[Y(t)]). \tag{2.19}$$

(c) At $t = T_{n-\frac{1}{2}}$*:* $\theta_n[Y(t)] = \pi$.

(d) When $T_{n-\frac{1}{2}} < t < T_n$*,* $\theta_n[Y(t)]$ *increases from* π *to* 2π *and is given by:*

$$\theta_n[Y(t)] = 2\pi - \cos^{-1}(\psi_n[Y(t)]). \tag{2.20}$$

(e) If needed, the process can be continued beyond T_n *by repeating the previous steps and adding* 2π *to the obtained values (and then,* 4π*,* $\cdots$*).*

3. **Functions governing the EOS condition:**
The functions $\{E_n\}$ governing the EOS condition of oscillatory problems are defined, for all n and in terms of the starting values $\{Y_{n-1}\}$, as:

$$\forall Y(t) \in \mathbb{R}^k, \quad E_n[Y(t)] = \theta_n[Y(t)] - 2\pi, \tag{2.21}$$

where $\theta_n(W)$ is evaluated, using the procedure given in definition 7.
The n^{th} slice is then ended when the solution completes a *full rotation* in the projection plane, i.e. as soon as the solution satisfies:

$$\begin{array}{c} \text{at } t = T_n,\ E_n[Y(T_n)] = 0, \quad \Longleftrightarrow \quad \theta_n[Y(T_n)] = 2\pi, \\ \text{with} \quad \theta_n[Y(t)] < 2\pi, \ \text{ if } T_{n-1} \leq t < T_n. \end{array} \tag{2.22}$$

The direct effect of this EOS condition is to make the polar angle φ_n, at the end of every slice, equal to the initial polar angle φ_0 corresponding to the initial value Y_0, yielding functions $\{\psi_n\}$ that depend all on the same initial value Y_0 of the solution, instead of the different starting values $\{Y_{n-1}\}$:

$$\psi_n[Y(t)] = \frac{\omega\vec{M}_{n-1}}{\left\|\omega\vec{M}_{n-1}\right\|_2} \cdot \frac{\omega\vec{M}(t)}{\left\|\omega\vec{M}(t)\right\|_2} = \frac{\omega\vec{M}_0}{\left\|\omega\vec{M}_0\right\|_2} \cdot \frac{\omega\vec{M}(t)}{\left\|\omega\vec{M}(t)\right\|_2} = \psi[Y(t)]. \tag{2.23}$$

The function $\{\psi_n(.)\}$, and therefore the functions $\{\theta_n(.)\}$ and $\{E_n(.)\}$, are independent of the starting value Y_{n-1}.
We let: $\forall n,\ \psi_n(.) = \psi(.)$, $\theta_n(.) = \theta(.)$ and $E_n(.) = E(.)$.
The expression of $\{E_n\}$, given in (2.21), simplify then to:

$$\forall Y(t) \in \mathbb{R}^k, \quad E[Y(t)] = \theta[Y(t)] - 2\pi. \tag{2.24}$$

and the EOS condition (2.22) simplifies also to:

$$\begin{array}{c} \text{at } t = T_n,\ E[Y(T_n)] = 0, \quad \Longleftrightarrow \quad \theta[Y(T_n)] = 2\pi, \\ \text{with} \quad \theta[Y(t)] < 2\pi, \ \text{ if } T_{n-1} < t < T_n. \end{array} \tag{2.25}$$

Note that, because of the oscillatory behavior of the solution $Y(.)$, this EOS condition is guaranteed to be reached.

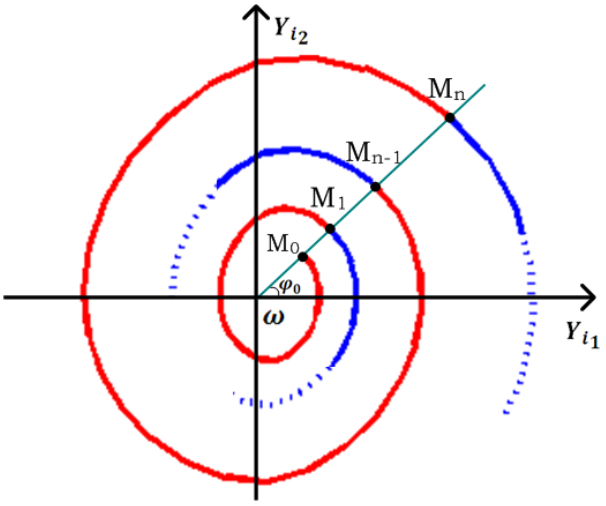


Figure 2.3: EOS Condition: Oscillating Problem

Particular periodic case:
In the case of a scalar second order problem, translating after lowering the order to a 2-dimensional system, a particular example would be the trivial periodic case that results on all slices in a common size ΔT and equal starting values $\{y_n\}$ and $\{y'_n\}$.

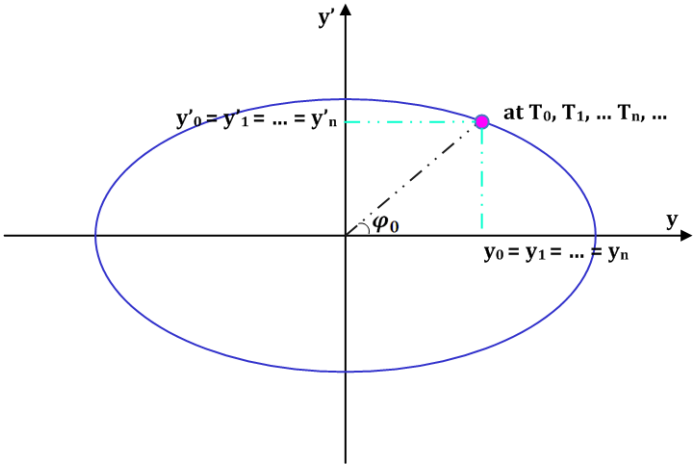


Figure 2.4: EOS Condition: Scalar Second Order Periodic Problem

4. **Rescaled EOS condition:**
If the nonzeroness condition (1.11) is satisfied, then the functions $\{H_n\}$, governing the rescaled EOS condition, have the invariance property according to definition 1, as proved below.
Under the nonzeroness condition (1.11) and after the change of variables (1.12), the expression (2.23) of $\psi[Y(t)]$ translates to:

$$\begin{aligned}\psi\left[D_n\left(\mathbf{1}+Z_n(s)\right)\right] &= \frac{Y_{0,i_1}Y_{n-1,i_1}\left(1+Z_{n,i_1}(s)\right)+Y_{0,i_2}Y_{n-1,i_2}\left(1+Z_{n,i_2}(s)\right)}{\sqrt{Y_{0,i_1}^2+Y_{0,i_2}^2}\sqrt{Y_{n-1,i_1}^2\left(1+Z_{n,i_1}(s)\right)^2+Y_{n-1,i_2}^2\left(1+Z_{n,i_2}(s)\right)^2}} \\ &= \frac{Y_{n-1,i_1}\left[Y_{0,i_1}\left(1+Z_{n,i_1}(s)\right)+Y_{0,i_2}\frac{Y_{n-1,i_2}}{Y_{n-1,i_1}}\left(1+Z_{n,i_2}(s)\right)\right]}{\left|Y_{n-1,i_1}\right|\sqrt{Y_{0,i_1}^2+Y_{0,i_2}^2}\sqrt{\left(1+Z_{n,i_1}(s)\right)^2+\frac{Y_{n-1,i_2}^2}{Y_{n-1,i_1}^2}\left(1+Z_{n,i_2}(s)\right)^2}}.\end{aligned}$$

But $\frac{Y_{n-1,i_1}}{|Y_{n-1,i_1}|} = \frac{Y_{0,i_1}}{|Y_{0,i_1}|}$, since all of the starting points $\{M_n\}_{n\geq 0}$ lie in the same quadrant.
Besides, the ratios $\left\{\frac{Y_{n,i_2}}{Y_{n,i_1}}\right\}$ are the expression of the tangent of the polar angles $\{\varphi_n\}$ at the starting points $\{M_n\}$ that are all equal to the initial polar angle φ_0, making for all n, $\frac{Y_{n,i_2}}{Y_{n,i_1}} = \tan\varphi_0$.
Hence, the function $\psi(.)$ translates, after rescaling and on all slices, to a function $\tilde{\psi}(.)$ that is independent of the starting values $\{Y_{n-1}\}$:

$$\begin{aligned}\tilde{\psi}\left[Z_n(s)\right] &= \psi\left[D_n\left(\mathbf{1}+Z_n(s)\right)\right] \\ &= \frac{Y_{0,i_1}}{|Y_{0,i_1}|}\frac{Y_{0,i_1}\left(1+Z_{n,i_1}(s)\right)+Y_{0,i_2}\tan\varphi_0\left(1+Z_{n,i_2}(s)\right)}{\sqrt{Y_{0,i_1}^2+Y_{0,i_2}^2}\sqrt{\left(1+Z_{n,i_1}(s)\right)^2+(\tan\varphi_0)^2\left(1+Z_{n,i_2}(s)\right)^2}}.\end{aligned}$$

This results in functions $\theta(.)$ and $E(.)$ translating also, after rescaling and on all slices, to functions $\tilde{\theta}(.)$ and $H(.)$ that are independent of n:

$$\begin{aligned}\tilde{\theta}\left[Z_n(s)\right] &= \theta\left[D_n\left(\mathbf{1}+Z_n(s)\right)\right] \\ H\left[Z_n(s)\right] &= E\left[D_n\left(\mathbf{1}+Z_n(s)\right)\right]\end{aligned} \tag{2.26}$$

Hence, a unique function $H:\mathbb{R}^k\to\mathbb{R}$ governs the rescaled EOS condition and is defined, independently of n, by:

$$\forall Z_n(s)\in\mathbb{R}^k,\ H\left[Z_n(s)\right] = \tilde{\theta}\left[Z_n(s)\right] - 2\pi. \tag{2.27}$$

Therefore, the n^{th} slice is ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{aligned}\text{at } s=s_n,\quad & H\left[Z_n(s_n)\right]=0 \iff \tilde{\theta}\left[Z_n(s_n)\right]=2\pi, \\ \text{with}\quad & \tilde{\theta}\left[Z_n(s)\right]<2\pi, \quad \text{if } 0<s<s_n.\end{aligned} \tag{2.28}$$

Remark 3 • *If, for some reason, the n^{th} time-slice is to be extended beyond T_n, the oscillatory behavior of the solution makes then: $|\theta\left[Y(t)\right]|>2\pi,\ \forall t>T_n$.*
This translates, after rescaling, to: $\forall s>s_n,\ \tilde{\theta}\left[Z_n(s)\right]>2\pi$.

- *In case of oscillatory problems in which the slices are ended when reaching the EOS condition (2.28), the assumption of boundedness of $\|Z_n(s)\|_\infty$ that is required by theorem 1 is not explicitly proved, but has been verified numerically.*

- *However, the additional assumption of the existence of a bijection, required in corollary 1 of theorem 1 is automatically satisfied when the time-slicing is done using EOS conditions (2.28), since it implies:*

$$\begin{aligned}&\text{at } s=s_n, && \tilde{\theta}\left[Z_n(s_n)\right]=2\pi, \\ &\text{if } 0<s<s_n, && \tilde{\theta}\left[Z_n(s)\right]<2\pi, \\ &\forall s>s_n, && \tilde{\theta}\left[Z_n(s)\right]>2\pi.\end{aligned}$$

making the function $H\left[Z_n(.)\right]$ define a bijection from $[0,\infty)$ toward $[-2\pi,\infty)$.

2.2.2 Alternative Choices

- When the EOS condition governed by the functions (2.21) is applied to oscillatory problems, a slice is ended whenever the trajectory of $M(t)$, in the i_1i_2-plane, hits the half-line that makes an angle equal to φ_0 with the i_1-axis.
A variant would be to end a slice whenever the trajectory of $M(t)$ hits the graph C,

lying in the quadrant containing the initial point M_0, of any continuous function Φ defined on all the semi-interval of $\mathbb{R}$ corresponding to this quadrant.

This is done in chapter 5, through an application to a membrane problem, and has the advantage of yielding (for a critical choice of $\{\beta_n\}$) an invariance property for a specific combination of the problem parameters.

- In some cases where the solution is known to pass regularly across the projection plane P (at every rotation), one could choose to end each n^{th} slice $[T_{n-1}, T_n]$ whenever the solution crosses the plane P, after completing a whole rotation.

  This EOS condition has been chosen in chapter 7, for solving a satellite orbit, in a non-Keplerian motion, that is known to be an osculating ellipse passing regularly across the Keplerian plane corresponding to the initial conditions. It has resulted in a stronger ratio property.

- In case the oscillatory problem has an explosive behavior and the only goal is to kill the stiffness of the problem, one could choose to use the EOS conditions described in section 1 of this chapter, for explosive problems.

  This has been done in [51] for solving a membrane problem, of which the solution explodes in an oscillatory way and in finite time, and have yielded (for a critical choice of $\{\beta_n\}$) a property of uniform similarity and accurate approximations for both the finite time of existence and the values of the solution.

2.3 Extension to Other Cases

2.3.1 Extinctive Problems

We consider, in this section, extinctive problems, satisfying:

$$\lim_{t\to\infty} \|Y(t)\|_\infty = 0,$$

of which the solution is known to keep, for $t > T_0$, a constant sign (for all of its components). Note first that, under this assumption and for $t > T_0$, the nonzeroness condition (1.11) is automatically satisfied.
The proposed EOS condition aim at limiting the decay of the $\|Y(t)\|_\infty$.

- **Case 1:** $\forall i,\ \forall t \geq 0,\quad Y_i(t)$ **has a constant sign**
  A possible choice of the functions $\{E_n\}$, governing the EOS condition, would be similar to that of the explosive case, defined in (2.6), by:

  $$E_n(W) = \left\|D_n^{-1}\left(W - Y_{n-1}\right)\right\|_\infty - S,$$

  but by letting:

  $$0 < S < 1. \tag{2.29}$$

  The EOS condition (1.2) acts then on the absolute value of the relative variation of each component of $Y(t)$ relatively to the same component of Y_{n-1} and prevents it from exceeding a cut-off value S.
  For a given $S \in (0,1)$, the n^{th} slice is ended as soon as the solution satisfies:

  $$\begin{array}{c} \text{at } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \left\|D_n^{-1}\left(Y(T_n) - Y_{n-1}\right)\right\|_\infty = S, \\ \text{with} \quad \left\|D_n^{-1}\left(Y(t) - Y_{n-1}\right)\right\|_\infty < S, \quad \text{if } T_{n-1} < t < T_n. \end{array} \tag{2.30}$$

  For all $S \in (0,1)$ and because of the continuity of the solution $Y(.)$ and its extinctive behavior, one has $E_n\left[Y(t)\right] = \left\|D_n^{-1}\left(Y(t) - Y_{n-1}\right)\right\|_\infty - S$ continuously varying from $E_n\left[Y(T_{n-1})\right] = -S < 0$ to $\lim_{t\to\infty} E_n\left[Y(t)\right] = 1 - S > 0$.
  This makes the set $\Delta_n = \{t > T_{n-1} \,|\, E_n\left[Y(t)\right] = 0\}$ not empty and therefore, the EOS condition (2.30) is guaranteed to be reached at $T_n = \min\{\Delta_n\}$.

  Particular Scalar Case:
  A particular case would be the scalar case, yielding: $\forall n,\ \left|\frac{Y_n - Y_{n-1}}{Y_{n-1}}\right| = S$, that translates, in case the solution $Y(t)$ has a constant sign and $|Y(t)|$ is decreasing, to:

  $$\forall n,\ \ 1 - \frac{Y_n}{Y_{n-1}} = S \qquad \iff \qquad \forall n,\ \ \frac{Y_n}{Y_{n-1}} = 1 - S = \text{ constant},$$

  making $\{Y_n\}$ a geometric sequence, with constant ratio equal to $0 < (1 - S) < 1$.

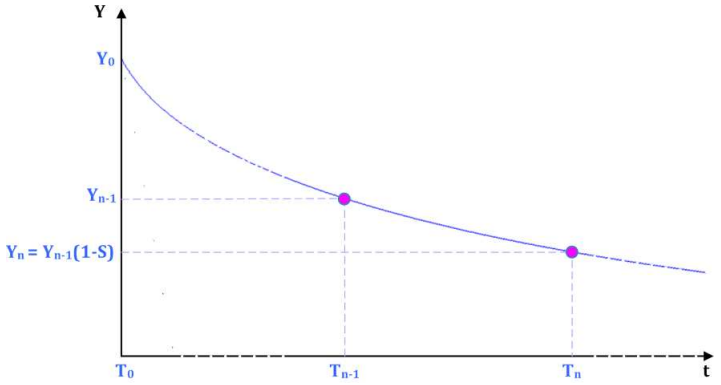


Figure 2.5: EOS Condition: Scalar Extinctive Positive Decreasing Problem

Rescaled EOS condition:
Rescaling $\{E_n\}$ yields functions $\{H_n\}$ having the invariance property $(\forall n, H_n = H)$, defined, in the same way as for the explosive problems, by (2.8):

$$\forall W \in \mathbb{R}^K,\ H(W) = \|W\|_\infty - S.$$

For a given cutoff value $S \in (0,1)$, the n^{th} slice is then ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{array}{lll} \text{at } s = s_n, & H\left[Z_n(s_n)\right] = 0 & \Longleftrightarrow \quad \|Z_n(s_n)\|_\infty = S, \\ & \text{with} \quad \|Z_n(s)\|_\infty < S, & \text{if } 0 < s < s_n. \end{array} \tag{2.31}$$

Remark 4 *If, moreover,* $||Y(.)||_\infty$ *monotonously decreases with* t*, one has:* $\forall t > T_n,\ \left\|D_n^{-1}\left(Y(t) - Y_{n-1}\right)\right\|_\infty - S > 0$ *and therefore,* $\forall s > s_n,\ \|Z_n(s)\|_\infty > S$.

- **Case 2:** $\exists T_0 > 0,\ \forall i,\ \forall t > T_0,\quad Y_i(t)$ **has a constant sign**
  In this case, the problem should first be solved on a preliminary time-slice that reaches an arbitrary time $T_1 > T_0$.
  From that point, the assumptions of case 1 are valid. The EOS condition (2.30) can then be used with any $S \in (0,1)$ for generating the next slices, as done in case 1.

2.3.2 Problems with an attraction point l

We consider, in this section, problems with an attraction point at infinity, satisfying:

$$\lim_{t\to\infty} Y(t) = l \neq 0 \in \mathbb{R}^K.$$

- **Case 1:** $\lim_{t\to\infty} Y(t) = l$ **is known**
  In this case, problem (S) is equivalent, through a simple preliminary change of variables $X = Y - l$, to the extinctive initial value problem in which one seeks the solution $X : [0,\infty) \to \mathbb{R}^K$, such that:

$$\left\{ \begin{array}{ll} \frac{dX}{dt} = F(X + l), & t > 0 \qquad (2.32.1) \\ X(0) = Y_0 - l. & \qquad (2.32.2) \end{array} \right. \tag{2.32}$$

If $X(t) = (Y(t) - l)$ is known to keep, for $t > T_0$, a constant sign (for all of its components), then the same EOS condition than that of extinctive problems, given in the previous section, can be used and applied on the variable X.

- **Case 2:** $\lim_{t\to\infty} Y(t) = l$ **is unknown**
One should first note that:

$$\lim_{t\to\infty} Y(t) = l \quad \Longleftrightarrow \quad \lim_{t\to\infty} \frac{dY}{dt} = 0 \quad \Longleftrightarrow \quad \lim_{t\to\infty} F(Y) = 0.$$

If the solution is such that $F(Y)$ is known to keep, for $t > T_0$, a constant sign (for all of its components), then the nonzeroness of $F(Y)$:

$$\forall i \in \{1, 2, \cdots, K\}, \;\; F_i(Y) \neq 0,$$

is automatically satisfied and an EOS condition could aim at limiting the decay of $\|F(Y)\|_\infty$.
In a first step, the problem should be solved on a preliminary time-slice that reaches an arbitrary time $T_1 > T_0$.
From that point, a possible choice of the functions $\{E_n\}$, governing the EOS condition, acts here on $F(Y)$ and is obtained by defining E_n, for all n, in terms of the starting value Y_{n-1}, as follows:

$$\forall W \in \mathbb{R}^K, \quad E_n(W) = \left\|D_{F_n}^{-1}\left[F(W) - F\left(Y_{n-1}\right)\right]\right\|_\infty - S, \tag{2.33}$$

with:

$$0 < S < 1, \tag{2.34}$$

D_{F_n} being the diagonal matrix associated with the vector $F\left(Y_{n-1}\right)$, which is invertible since $\forall i$, $F_i(Y) \neq 0$. For a given $S \in (0, 1)$, the n^{th} slice is then ended as soon as the solution satisfies:

$$\begin{array}{c} \text{at } t = T_n, \; E_n\left[Y\left(T_n\right)\right] = 0 \iff \left\|D_{F_n}^{-1}\left[F\left(Y\left(T_n\right)\right) - F\left(Y_{n-1}\right)\right]\right\|_\infty = S \\ \text{with} \quad \left\|D_{F_n}^{-1}\left[F\left(Y(t)\right) - F\left(Y_{n-1}\right)\right]\right\|_\infty < S, \quad \text{if } \; T_{n-1} < t < T_n. \end{array} \tag{2.35}$$

For all $S \in (0, 1)$ and because of the continuity of the solution $Y(.)$ and the extinctive behavior of its derivative at infinity, one has:

$$E_n\left[Y(t)\right] = \left\|D_{F_n}^{-1}\left[F\left(Y(t)\right) - F\left(Y_{n-1}\right)\right]\right\|_\infty - S,$$

that continuously vary from $E_n\left[Y(T_{n-1})\right] = -S < 0$ to $\lim_{t\to\infty} E_n\left[Y(t)\right] = (1-S) > 0$.
This makes the set $\Delta_n = \{t > T_{n-1} \,|\, E_n\left[Y(t)\right] = 0\}$ not empty and therefore, the EOS condition (2.35) guaranteed to be reached at $T_n = \min\{\Delta_n\}$.

Rescaled EOS condition:
Applied to $Y(t)$ $(t > T_{n-1})$, the function E_n defined in (2.33) is rescaled as:

$$E_n\left[Y_{n-1} + D_n Z_n(s)\right] = \left\|D_{F_n}^{-1}\left[F\left(Y_{n-1} + D_n Z_n(s)\right) - F\left(Y_{n-1}\right)\right]\right\|_\infty - S.$$

This yields a family $\{H_n\}$ of functions, governing the rescaled EOS condition, given

by:

$$\forall W \in \mathbb{R}^K,\ H_n(W) = \left\| D_{F_n}^{-1} \left[F\left(Y_{n-1} + D_n W\right) - F\left(Y_{n-1}\right) \right] \right\|_\infty - S. \tag{2.36}$$

Clearly, the functions $\{H_n\}$ are parametrized by the corresponding starting values $\{Y_{n-1}\}$ of each slice. Therefore, they are not invariant according to definition 1.
For a given cutoff value $S > 0$, the n^{th} slice is ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{array}{rl} & \text{at } s = s_n, \quad H_n\left[Z_n(s_n)\right] = 0 \iff \\ & \left\| D_{F_n}^{-1} \left[F\left(Y_{n-1} + D_n Z_n(s_n)\right) - F\left(Y_{n-1}\right) \right] \right\|_\infty = S, \\ \text{with} & \left\| D_{F_n}^{-1} \left[F\left(Y_{n-1} + D_n Z_n(s)\right) - F\left(Y_{n-1}\right) \right] \right\|_\infty < S, \quad \text{if } 0 < s < s_n. \end{array} \tag{2.37}$$

- **Another way for ending the time slices**
When the solution of (S) is known to have an attraction point l (that could be zero), a different EOS condition can be used, as partially done in [53].
Unlike all previous choices, this EOS condition is not governed by a family of functions: it rather consists in the choice of a constant number n_0 of rescaled time-steps on which the rescaled solution should be advanced on every time-slice.
However, the resulting coarse grid is not necessarily regular.
The size s_n of each rescaled time-slice is constant:

$$\forall n, \quad s_n = s_0 = n_0 \tau_0,$$

where τ_0 is the constant size of each rescaled time-step. But the size of each time-slice of the coarse grid is modulated by the time-rescaling factor β_n:

$$\forall n, \quad T_n - T_{n-1} = \beta_n s_0.$$

The choice $\beta_n = 1\,(\forall n)$ is the only one yielding a regular coarse grid.

The Initial Value Problem (S) is equivalent here to a sequence of rescaled Initial Value Problems in which one seeks for the rescaled function $Z_n : [0, s_0] \to \mathbb{R}^K$, on each n^{th} slice corresponding to $[T_{n-1}, T_n]$, such that:

$$\begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_0, \quad (2.38.1) \\ Z_n(0) = 0, & \quad (2.38.2) \end{cases} \tag{2.38}$$

where

$$G_n(Z_n) = \beta_n D_n^{-1} F\big(Y_{n-1} + D_n Z_n\big).$$

Note that since $\lim_{t\to\infty} Y(t) = l \in \mathbb{R}^K$, one deduces: $\lim_{n\to\infty} Y_n = l$ and $\lim_{n\to\infty} D_n = D_l$, where D_l is the diagonal matrix associated with $l \in \mathbb{R}^K$.
Together with the end-of-slice relation $Y_n = Y_{n-1} + D_n Z_n(s_n)$, this implies:

$$\lim_{n\to\infty} Z_n(s_n) = 0. \tag{2.39}$$

Chapter 3

Case of an Algebraic Function

For solving the initial value problem (S) : $\frac{dY}{dt} = F(Y),\ Y(0) = Y_0$, we have defined, in Chapter 1, an equivalent sequence of initial value shooting problems, on the basis of an End-Of-Slice condition that should be reached, and under the assumption of existence of such EOS condition. In Chapter 2, we have proved the existence of EOS conditions guarantying the generation of a coarse grid and chosen appropriately, depending on the behavior of the solution (explosive, oscillatory,...).
In this Chapter, we consider a study case where the components of $F(.)$ have the algebraic form:

$$(F(Y))_i = \sum_j a_{ij}(Y_i)^{k_{ij}}(Y_j)^{l_{ij}}, \tag{3.1}$$

such that $a_{ij} \in \mathbb{R}$ and the coefficients k_{ij} and l_{ij} are nonnegative integers assumed to satisfy:

$$\forall i, \forall j, \qquad k_{ij} \geq 1, \quad l_{ij} \geq 1, \tag{3.2}$$

in order to have F satisfying Lipschitz properties.

In section 1, we prove the existence of an EOS condition generating, in a unique way, an increasing sequence $\{T_n\}_{n\geq 0}$ satisfying (1.1): $\cup_{n\geq 1}[T_{n-1}, T_n] = [0, T]_{T\leq\infty}$. Then, we show how, from such existence analysis, could derive a related sequence $\{T'_n\}$ and a related EOS condition making problem (S) equivalent to an infinite sequence of initial value shooting problems (S_n). It should be noticed that this is done, regardless of the behavior of the solution.
In section 2, we explain, on this study case, how a "critical" choice for the time-rescaling parameters $\{\beta_n\}$ can be done in a way to yield a uniform similarity.

On démontre, dans ce cas et pour une suite positive $\{\rho_n\}$ donnée, qu'il existe une suite $\{(T_n, Y_n)\}_{n=1,2,\dots} \subset (0,\infty)\times\mathbb{R}^K$, où $Y_n = Y(T_n)$, déduite de $(T_0 = 0, Y_0)$ et $\{\rho_n\}$, telle que $\{T_n\}$ est croissante et le problème (S) a une solution unique $Y : [0, T_m] \to \mathbb{R}^K$ vérifiant : $\forall n \in 1, 2, \cdots, m,\ \forall t \in [T_{n-1}, T_n]\,, \quad \|Y(t) - Y_{n-1}\|_\infty \leq \rho_n$.
On montre ensuite comment, de cette étude d'existence et d'unicité de la solution, on peut déduire l'existence d'une suite croissante $\{T'_n\}$ et d'une suite positive $\{\rho'_n\}$, avec $\forall n,\ \rho'_n \leq \rho_n$ et $T'_n \leq T_n$, qui rendent la résolution du problème (S) équivalente à celle d'une suite de problèmes (S_n) de tirs à valeur initiale dans lesquels la condition EOS est

donnée par la fonction:

$$E_n\left(Y(t)\right) = \|Y(t) - Y_{n-1}\|_\infty - \rho'_n,$$

et s'écrit :

$$\begin{array}{c} \text{à } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \|Y\left(T_n\right) - Y_{n-1}\|_\infty = \rho'_n, \\ \text{avec} \quad \|(Y(t) - Y_{n-1})\|_\infty < \rho'_n, \quad \text{si } \ T_{n-1} < t < T_n. \end{array}$$

L'intérêt de cette étude est de prouver l'existence d'une condition EOS générant une grille grossière, indépendemment du comportement de la solution !

In section 3, we show how the invariance and asymptotic similarity of the rescaled initial value shooting problems express, in this study case .
Note that the analysis done in the 3 first sections keeps holding, in the same way, when $F(Y) = F_1(Y) + F_2(Y) + ...$, where each of $F_1,\ F_2,\ ...$ is of the form (3.1). However, a single term is considered in this chapter, for a simplicity purpose.
In section 4, we consider the particular linear case of F, when $\forall i,\ \forall j\ \ k_{ij} = 0$ and $l_{ij} = 1$, and we prove the resulting uniform and asymptotic similarity of the rescaled problems, for some choices of $\{\beta_n\}$ and under some conditions.

3.1 Proof of the Existence of an EOS Condition defining a Coarse Grid

The procedure is mainly based on a local approach, as that used for proving the existence and uniqueness of a solution to an initial value problem.

For all $\rho > 0$ and for all $\omega \in \mathbb{R}^K$, the closed hyperball $B_\rho(\omega)$ of $\mathbb{R}^K$ is defined using the infinity norm:

$$B_\rho[\omega] = \left\{Y \in \mathbb{R}^K,\ \|Y - \omega\|_\infty \leq \rho\right\}. \tag{3.3}$$

When F is of the form (3.1), assumption (3.2) allows to define $F_Y(Y),\ \forall Y \in \mathbb{R}^K$:

$$\left(F_Y(Y)\right)_{im} = \left\{ \begin{array}{ll} \sum_j a_{ij}(Y_i)^{k_{ij}-1}(Y_j)^{l_{ij}}, & \text{if } m = i \\ a_{im}(Y_i)^{k_{im}}(Y_m)^{l_{im}-1}, & \text{if } m \neq i \end{array} \right. ,$$

and to prove consequently the following results.

Lemma 1 *Given a function F of the form (3.1) satisfying assumption (3.2), and given any constant $\rho > 0$ and any vector $\omega \in \mathbb{R}^K$, there exist $M, L \in \mathbb{R}^+$, depending on (ρ, ω), such that:*

$$\forall X \in B_\rho[\omega], \qquad \left\{ \begin{array}{l} \|F(X)\|_\infty \leq M, \\ \|F_Y(X)\|_\infty \leq L. \end{array} \right. \tag{3.4}$$

Proof.
Because of the continuity of F and $F_Y : \mathbb{R}^K \to \mathbb{R}^K$ and that of the norm $\|.\|_\infty : \mathbb{R}^K \to \mathbb{R}$, and because $B_R(\omega)$ is a compact space of $\mathbb{R}^K$, the boundedness theorem states that there exist positive numbers M and L such that: $\forall X \in B_\rho[\omega],\ \|F(X)\|_\infty \leq M$ and $\|F_Y(X)\|_\infty \leq L$. The extreme value theorem guarantees then that the maximum values are reached. Hence:
$M = \sup_{X \in B_\rho[\omega]} \|F(X)\|_\infty \quad$ and $\quad L = \sup_{X \in B_\rho[\omega]} \|F_Y(X)\|_\infty$. ∎

Theorem 4 *Given the Initial Value Problem (S), where F is of the form (3.1) and satisfies assumption (3.2),*
and given a sequence $\{\rho_n\}_{n=1,2,\dots}$ of positive numbers,
there exists a sequence $\{(T_n, Y_n)\}_{n=1,2,\dots} \subset (0,\infty)\times\mathbb{R}^K$, with $Y_n = Y(T_n)$, obtained from $(T_0 = 0, Y_0)$ and $\{\rho_n\}$, satisfying:

1. *$\{T_n\}$ is increasing: $0 < T_1 < T_2 < \cdots < T_n < \cdots$, with $\Delta T_n = T_n - T_{n-1}$ depending on the choice of $\{\rho_n\}$,*
2. *For all $m = 1, 2, \cdots$, problem (S) has a unique solution $Y : [0, T_m] \to \mathbb{R}^K$ satisfying:*

$$\forall n \in 1, 2, \cdots, m,\ \ \forall t \in [T_{n-1}, T_n], \quad Y(t) \in B_{\rho_n}[Y_{n-1}], \tag{3.5}$$

where $B_{\rho_n}[Y_{n-1}] = \left\{V \in \mathbb{R}^K,\ \ \|V - Y_{n-1}\|_\infty \le \rho_n\right\}$.

Proof.
The proof uses mathematical induction.
Base step: Existence of a time $T_1 > 0$, satisfying (3.5).
The Volterra integral equation that is equivalent to problem (S) is:

$$Y(t) = Y_0 + \int_0^t F(Y(s))ds.$$

For $T_1 > 0$, consider the set of continuous functions:

$$C_1 = \left\{W : [0, T_1] \to \mathbb{R}^K \mid \forall s \in [0, T_1],\ W(s) \in B_{\rho_1}[Y_0]\right\},$$

on which one can define the integral operator Π_1:

$$\Pi_1 : \quad W(s) \longmapsto X(t) = \Pi_1\left(W(t)\right) = Y_0 + \int_0^t F(W(s))ds.$$

Using lemma (1) and given ρ_1, there exist constants $M_1, L_1 \in \mathbb{R}$, depending on (ρ_1, Y_0), such that for all $X \in B_{\rho_1}[Y_0]$, one has:

$$\begin{cases} \|F(X)\|_\infty \le M_1 \\ \|F_Y(X)\|_\infty \le L_1 \end{cases}.$$

It follows that $\|X(t) - Y_0\| \le \int_0^t \|F(W(s))\|\, ds \le M_1 t$, implying $M_1 t \le \rho_1$ (for $\Pi(W(.))$ to be in C_1).
Besides, $\forall W_1, W_2 \in C_1, X_1(t) - X_2(t) = \int_0^t \left[F(W_1(s)) - F(W_2(s))\right] ds$.
Therefore: $\|X_1(t) - X_2(t)\| \le \int_0^t \|F_Y(\xi(s))\|\, \|W_1(s) - W_2(s)\|\, ds$, where $\xi \in C_1$,
implying:

$$\max_{0\le s\le t}\left\{\|X_1(s) - X_2(s)\|\right\} \le L_1 t \max_{0\le s\le t}\left\{\|W_1(s) - W_2(s)\|\right\}.$$

For the integral operator Π to have a unique fixed point, it should be a contracting map (Banach fixed point theorem), implying $L_1 t < 1$. Together with $M_1 t < \rho_1$, this condition yields a choice $T_1 = \min\left\{\frac{\rho_1}{M_1}, \frac{1}{L_1+\epsilon}\right\} > 0$, where ϵ is an arbitrary small number, and therefore a first time-slice $[0, T_1]$, on which Picard-Lindelöf's theorem guarantees the existence and uniqueness of a solution $Y(.)$ to problem (S), this solution satisfying: $\forall t \in [0, T_1], \quad Y(t) \in B_{\rho_1}[Y_0]$.

Inductive step: The existence of a time T_{n-1} satisfying (3.5) implies the existence of a time $T_n > T_{n-1}$, satisfying (3.5).
The same argument is repeated on:

$$Y(t) = Y_{n-1} + \int_{T_{n-1}}^{T_{n-1}+t} F(Y(s))ds,$$

after defining, for $T_n > T_{n-1}$, and on the set of continuous functions:

$$C_n = \left\{ W : [T_{n-1}, T_n] \to \mathbb{R}^K \mid \forall s \in [T_{n-1}, T_n],\ W(s) \in B_{\rho_n}[Y_{n-1}] \right\},$$

the integral operator Π_n:

$$\Pi_n : \quad W(s) \longmapsto X(t) = \Pi_n(W)(t) = Y_0 + \int_0^t F(W(s))ds.$$

Using lemma (1) and given ρ_n, there exist constants $M_n, L_n \in \mathbb{R}$, depending on (ρ_n, Y_{n-1}), such that for all $X \in B_{\rho_n}[Y_{n-1}]$, one has:

$$\begin{cases} \|F(X)\|_\infty \leq M_n \\ \|F_Y(X)\|_\infty \leq L_n \end{cases}.$$

In the same way, one can obtain a choice of T_n such that $T_n - T_{n-1} = \min\left\{ \frac{\rho_n}{M_n}, \frac{1}{L_n+\epsilon} \right\}$ and define an n^{th} time-slice $[T_{n-1}, T_n]$ on which Picard-Lindelöf's theorem guarantees the existence and uniqueness of a solution $Y(.)$ to problem (S). This solution satisfies: $\forall t \in [T_{n-1}, T_n], \quad Y(t) \in B_{\rho_n}[Y_{n-1}]$.
The two steps prove (3.5), for all n. ∎

Corollary 3
Under the assumptions of theorem (4), there exist a sequence $\{T'_n\}$ and a sequence $\{\rho'_n\}$, satisfying $\forall n$, $\rho'_n \leq \rho_n$ and $T'_n \leq T_n$, and making problem (S) equivalent to a sequence of initial value shooting problems in which one seeks, on each n^{th} slice, a time T'_n and a solution $Y : [T'_{n-1}, T'_n] \to \mathbb{R}^k$ satisfying:

$$(S_n) \quad \begin{cases} \frac{dY}{dt} = F(Y), & T_{n-1} < t \leq T_n & (3.6.1) \\ Y(T'_{n-1}) = Y_{n-1}, & & (3.6.2) \\ E_n(Y(T_n)) = 0 \quad \textit{and} \quad \forall t \in (T_{n-1}, T_n), \ E_n(Y(t)) \neq 0, & & (3.6.3) \end{cases} \tag{3.6}$$

where $E_n(Y(t)) = \|Y(t) - Y_{n-1}\|_\infty - \rho'_n$.

Proof.
Let $T'_0 = T_0 = 0$.
The proof is obtained using the following steps, for all $n \geq 1$:

- let $\rho'_n = \sup_{t \in [T'_{n-1}, T_n]} \|Y(t) - Y_{n-1}\|_\infty$,
- find $T'_n \in (T'n - 1, T_n]$, the first time of the interval at which $\|Y(t) - Y_{n-1}\|_\infty = \rho'_n$.
- update T_n by the new starting time T'_n and obtain T_{n-1} accordingly. ∎

3.2 Existence of $\{\beta_n\}$ Yielding Uniform Similarity

After selection of an EOS condition and rescaling, problem (S) is equivalent to the rescaled systems (S'_n) given in (1.18). When applied to the study case, the functions $G_n(Z_n) = \beta_n D_n^{-1} F\big(Y_{n-1} + D_n Z_n\big)$ are component-wise given by:
$(G_n(Z_n))_i = \beta_n \alpha_{n,i}^{-1} \sum_j a_{ij} \left(Y_{n-1,i} + \alpha_{n,i} Z_{n,i}\right)^{k_{ij}} \left(Y_{n-1,j} + \alpha_{n,j} Z_{n,j}\right)^{l_{ij}}$,
or equivalently:
$(G_n(Z_n))_i = \beta_n \alpha_{n,i}^{-1} \sum_j a_{ij} (\alpha_{n,i})^{k_{ij}} (\alpha_{n,j})^{l_{ij}} \left(\frac{Y_{n-1,i}}{\alpha_{n,i}} + Z_{n,i}\right)^{k_{ij}} \left(\frac{Y_{n-1,j}}{\alpha_{n,j}} + Z_{n,j}\right)^{l_{ij}}$.
Using the explicit expression (1.6) for α_n, we let for all i,

$$\delta_{n,i} = \frac{Y_{n-1,i}}{\alpha_{n,i}} = \begin{cases} 1, & \text{if } Y_{n-1,i} \neq 0 \\ 0, & \text{if } Y_{n-1,i} = 0 \end{cases}.$$

It follows:
$(G_n(Z_n))_i = \beta_n \sum_j a_{ij} (\alpha_{n,i})^{k_{ij}-1} (\alpha_{n,j})^{l_{ij}} \left(\delta_{n,i} + Z_{n,i}\right)^{k_{ij}} \left(\delta_{n,j} + Z_{n,j}\right)^{l_{ij}}$,
or equivalently:

$$(G_n(Z_n))_i = \sum_j \gamma_{n,ij} \left(\delta_{n,i} + Z_{n,i}\right)^{k_{ij}} \left(\delta_{n,j} + Z_{n,j}\right)^{l_{ij}}, \tag{3.7}$$

where $\gamma_{n,ij} = \beta_n a_{ij} (\alpha_{n,i})^{k_{ij}-1} (\alpha_{n,j})^{l_{ij}}$.
The Jacobian of G_n can also be derived. Its general form is:

$$\begin{aligned} [J_{G_n}(Z_n)]_{im} &= \tfrac{\partial (G_n(Z_n))_i}{\partial Z_{n,m}} \\ &= \begin{cases} \gamma_{n,im} \left(\delta_{n,i} + Z_{n,i}\right)^{k_{im}} \left(\delta_{n,m} + Z_{n,m}\right)^{l_{im}-1} & \text{if } m \neq i \\ \sum_j \left\{ \gamma_{n,ij} \left(\delta_{n,i} + Z_{n,i}\right)^{k_{ij}-1} \left(\delta_{n,j} + Z_{n,j}\right)^{l_{ij}} \right\} & \text{if } m = i. \end{cases} \end{aligned} \tag{3.8}$$

β_n is selected such that:

$$\begin{cases} \forall n, \ \forall i, \ \forall j, \quad |\gamma_{n,ij}| = \beta_n \left|a_{ij}\right| \left|\alpha_{n,i}\right|^{k_{ij}-1} \left|\alpha_{n,j}\right|^{l_{ij}} \leq 1, \\ \exists i_0, \ \exists j_0, \quad \forall n \quad |\gamma_{n,i_0 j_0}| > 0. \end{cases}$$

The first condition is meant for controlling the problem parameters and the second prevents G_n from being equal to the trivial 0-function.
This leads to a possible **critical** sequence $\{\beta_n\}$ given, for all n, by:

$$\beta_n = \frac{1}{\max_{i,j} \left\{ \left|a_{ij}\right| \left|\alpha_{n,i}\right|^{k_{ij}-1} \left|\alpha_{n,j}\right|^{l_{ij}} \right\}}. \tag{3.9}$$

The resulting parameters $\gamma_{n,ij}$, satisfying $|\gamma_{n,ij}| \leq 1$ with at least one of these coefficients equal to 1, give rise, on each n^{th} slice, to a critical function G_n.

Proposition 2
If the initial value problem $\frac{dY}{dt} = F(Y), \ Y(0) = Y_0$, with F having the algebraic form (3.1) and satisfying (3.2), is rescaled using the critical choice (3.9) for $\{\beta_n\}$, then the rescaled initial value problems:

$$(IVP_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

are uniformly similar.

Proof.
From the critical choice (3.9), It follows directly from (3.7), (3.8) and (3.9) that if $||Z_n(s)||_\infty \leq S\,(\forall s)$, then:
$\sup_{||Z_n||_\infty \leq S} \{\|G_n(Z_n)\|_\infty\} \leq \max_i \sum_j (1+S)^{k_{ij}} (1+S)^{l_{ij}} = C$ and
$\sup_{||Z_n||_\infty \leq S} \{\|J_{G_n}(Z_n)\|_\infty\} \leq \max_i \left\{\sum_j \left[(1+S)^{k_{ij}-1} (1+S)^{l_{ij}}\right]\right.$

$$\left. + \sum_{m \neq i} \left[(1+S)^{k_{im}} (1+S)^{l_{im}-1}\right]\right\} = C',$$

where C and C' are independent from n. ■

3.3 Occurrence of Invariance and Asymptotic Similarity

Proposition 3 ***: Invariance***
If the functions $\{H_n\}$ governing the rescaled EOS condition are invariant,
if also $\forall n,\ \forall i,\ \ \delta_{n,i} = \delta_{1,i}$, i.e. the i^{th} component of Y_n is either zero or nonzero, independently of n,
and if $\forall n,\ \forall i,\ \forall j,\ \ \gamma_{n,ij} = \gamma_{1,ij}$, i.e. all the coefficients $\gamma_{n,ij}$ are independent of n, then the rescaled systems:

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), & 0 < s \leq s_n, \\ Z_n(0) = 0, & \\ H\left[Z_n(s)\right] \neq 0,\ \ \forall s < s_n \text{ and} & H_n\left[Z_n(s_n)\right] = 0, \end{cases}$$

are invariant.

Proof.
From the particular form (3.1) of F follows the explicit expression (3.7) for G_n:

$$\left(G_n(Z_n)\right)_i = \sum_j \gamma_{n,ij} \left(\delta_{n,i} + Z_{n,i}\right)^{k_{ij}} \left(\delta_{n,j} + Z_{n,j}\right)^{l_{ij}},$$

If $\forall n,\ \forall i,\ \forall j,\ \ \gamma_{n,ij} = \gamma_{1,ij}$, and since $\forall n,\ \forall i,\ \ \delta_{n,i} = \delta_{1,i}$, then:

$$\forall n,\ \forall W \in \mathbb{R}^K\ \forall i, \quad \left(G_n(W)\right)_i = \left(G_1(W)\right)_i,$$

yielding $\forall n,\ \ G_n(.) = G_1(.)$. Hence the invariance of the rescaled systems. ■

Proposition 4 ***: Asymptotic similarity***
If the functions $\{H_n\}$ governing the rescaled EOS condition are invariant,
if also the rescaled initial value problems (S'_n) associated with the initial value problem (S), with F having the algebraic form (3.1) and satisfying (3.2), are such that there exists $S > 0$ satisfying:

$$\forall n,\ \ \forall s \in [0, s_n], \quad Z_n(s) \in B_S = \left\{W \in \mathbb{R}^K,\ \ \|W\|_\infty \leq S\right\},$$

and if $\forall i,\ \forall j,\ \ \lim_{n\to\infty}(\gamma_{n,ij}) = \gamma_{L,ij}$, i.e. all the sequences of coefficients $\{\gamma_{n,ij}\}$ are

convergent,
then the rescaled systems (S'_n) *are asymptotically similar to the limit system defined by the function* G_L *obtained by substituting, in the expression of* G_n*, the coefficients* $\{\gamma_{L,ij}\}$ *for* $\{\gamma_{n,ij}\}$*.*

Proof.
If $\forall i,\ \forall j,\quad \lim_{n\to\infty}(\gamma_{n,ij}) = \gamma_{L,ij}$, then:
$\forall \epsilon,\ \forall i,\ \forall j,\quad \exists \eta(i,j,\epsilon),\quad \forall n \geq \eta,\quad |\gamma_{n,ij} - \gamma_{L,ij}| < \epsilon$.
By letting $n_0(\epsilon) = \max_{i,j}\eta(i,j,\epsilon)$, it follows:
$\forall \epsilon,\quad \exists n_0(\epsilon),\quad \forall n \geq n_0,\ \forall i,\ \forall j,\quad |\gamma_{n,ij} - \gamma_{L,ij}| < \epsilon$.
By defining, component-wise and for all $W \in \mathbb{R}^K$, $G_L(W)$ by:
$(G_L(W))_i = \sum_j \gamma_{L,ij}\,(\delta_{n,i} + W_i)^{k_{ij}}\,(\delta_{n,j} + W_j)^{l_{ij}}$, one gets:
$\forall W,\ \forall i,\quad [G_n(W) - G_L(W)]_i = \sum_j (\gamma_{n,ij} - \gamma_{L,ij})\,(\delta_{n,i} + W_i)^{k_{ij}}\,(\delta_{n,j} + W_j)^{l_{ij}}$.
It follows: $\forall \epsilon,\quad \exists n_0(\epsilon),\quad \forall n \geq n_0,\ \forall W \in B_S,\quad \|G_n(W) - G_L(W)\|_\infty < \epsilon'$,
where $\epsilon' = \left\{\max_i \sum_j \left[\epsilon(1+S)^{k_{ij}+l_{ij}}\right]\right\}$.
Hence, the uniform convergence of $G_n(.)$ toward $G_L(.)$ and the asymptotic similarity of the rescaled models to the limit system defined by G_L. ■

3.4 Linear Case

Consider the first order initial value problem in which one seeks $Y : [0, T] \to \mathbb{R}^k$, such that:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = AY, & t > 0, \quad (3.10.1) \\ Y(0) = Y_0 > 0. & \quad (3.10.2) \end{cases} \tag{3.10}$$

where A is a constant matrix in $\mathbb{R}^{K\times K}$.
Note that this linear form of $F(Y)$ is a particular case of the algebraic form (ref) studied in this chapter, with $\forall i,\ \forall j \quad k_{ij} = 0$ and $l_{ij} = 1$.
Note that, in this linear case, F does not satisfy the condition (3.2).

3.4.1 Analytic Solution

The analytic solution of this problem is known to be: $Y(t) = e^{At}Y_0$, the matrix exponential being defined, for all $K \times K$ matrix X, by $e^X = exp(X) = \sum_{j=0}^{\infty} \frac{1}{j!}X^j$.

Proposition 5 *Assume that the eigenvalue λ of A having the largest real part is unique (therefore, this eigenvalue is real since $A \in \mathbb{R}^{K\times K}$) and let u be a normalized eigenvector corresponding to λ.*
Then, for all $x_0 \in \mathbb{R}^K$ satisfying $u^*Ax_0 \neq 0$:

$$\lim_{t\to\infty} \frac{e^{tA}x_0}{\|e^{tA}x_0\|_2} = \pm u \tag{3.11}$$

Proof:
Consider a Schur decomposition of A:

$$A = QTQ^*.$$

Q is a unitary matrix (so that its inverse Q^{-1} is also conjugate transpose Q^* of Q), and its columns are orthonormal eigenvectors of A:

$$Q = [u_1, u_2, \cdots, u_k] \qquad \text{with} \quad u_1 = u.$$

T is an upper triangular matrix (Schur form of A). Since T is similar to A, it has the same multiset $\{\lambda_1, \lambda_2, \cdots, \lambda_k\}$ of eigenvalues, with $\lambda_1 = \lambda$, and since it is triangular, those eigenvalues are the diagonal entries of T.
The analytic solution of problem (3.10) can then be expressed as:

$$Y(t) = e^{tA}Y_0 = Qe^{tT}Q^*Y_0, \tag{3.12}$$

where the diagonal entries of the upper triangular matrix e^{tT} are $\{e^{t\lambda_1}, e^{t\lambda_2}, \cdots, e^{t\lambda_k}\}$, with $e^{t\lambda_1} = e^{t\lambda}$.
A block decomposition of the upper triangular matrix T, emphasizing on the eigenvalue λ, would be:

$$T = \left(\begin{array}{c|c} \lambda & r^* \\ \hline 0 & S \end{array}\right) \tag{3.13}$$

It follows that:

$$e^{tA} = Qe^{tT}Q^* = Q\left(\begin{array}{c|c} e^{t\lambda} & s_t^* \\ \hline 0 & e^{tS} \end{array}\right)Q^*,$$

where s_t^* is the solution of the Sylvester equation derived from the identity $Te^{tT} = e^{tT}T$ (having the same invariant subspaces, T and e^{tT} commute) that yields: $\lambda s_t^* + r^*e^{tS} = e^{t\lambda}r^* + s_t^*S$, or equivalently:

$$s_t^*\left(\lambda I - S\right) = e^{t\lambda}r^* - r^*e^{tS}. \tag{3.14}$$

Let $\Lambda(S)$ be the spectrum of S. From the assumptions of the proposition, one has: $\forall\mu \in \Lambda(S),\ \lambda > Re(\mu)$, with $\lambda \notin \Lambda(S)$; it follows that $(\lambda I - S)$ is invertible and the Sylvester equation (3.14) has a unique solution s_t^*.

Let $x_t = e^{tA}x_0$ and $y_t = e^{-t\lambda}e^{tA}x_0 = e^{t(A-\lambda I)}x_0$.
As $t \to \infty$, the limit of x_t can be deduced from that of y_t,as shown below.

One has: $y_t = Q\left(\begin{array}{c|c} 1 & e^{-\lambda t}s_t^* \\ \hline 0 & e^{t(S-\lambda I)} \end{array}\right)Q^*x_0$ with all eigenvalues of $(S - \lambda I)$ having negative real parts. It follows that $\lim_{t\to\infty} e^{t(S-\lambda I)} = 0$.

Let now: $w_t = e^{-\lambda t}s_t$.
This vector is a solution for the following equation, deduced from (3.14):

$$\lambda w_t^* - w_t^*S = r^* - r^*e^{t(S-\lambda I)}. \tag{3.15}$$

Since $\lim_{t\to\infty} e^{t(S-\lambda I)} = 0$, equation (3.15) reduces, as $t \to \infty$, to the following limit equation:

$$\lambda w^* - w^*S = r^* \qquad \Longleftrightarrow \qquad \lambda w - S^*w = r,$$

proving that:

$$\lim_{t\to\infty} w_t = w \qquad \text{with} \qquad w = \left(\lambda I - S^*\right)^{-1} r, \tag{3.16}$$

and yielding:

$$\lim_{t\to\infty} y_t = Q\left(\begin{array}{c|c} 1 & w^* \\ \hline 0 & 0 \end{array}\right)Q^*x_0 = Q\begin{pmatrix} \mu \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where $\mu = \begin{pmatrix} 1 & w^* \end{pmatrix} Q^*x_0$ is the limit, when $t \to \infty$, of $\mu_t = u_t^*e^{t(A-\lambda I)}x_0$.

General case: $\mu \neq 0$
Having in this case $\lim_{t\to\infty} y_t = \mu u$, and since $\frac{x_t}{\|x_t\|_2} = \frac{y_t}{\|y_t\|_2}$, it follows that:

$$\lim_{t\to\infty} \frac{x_t}{\|x_t\|_2} = \frac{\mu}{|\mu|}u = \pm u.$$

Particular case: $\mu = 0$
This happens in two ways:
(i) μ_t is the zero-function, implying $u_1^*Ax_0 = 0$, contradicting the assumptions.
(ii) $\mu_t = 0$ for a finite number of values t_i (including $t_j = \infty$), implying:
$\exists t_m,\ \forall t > t_m,\ \mu_t \neq 0$.

It follows that u and $-u$ are the two only possible accumulation points of the sequence $\{\frac{x_t}{\|x_t\|_2}\}$. ■

Corollary 4 *If the eigenvalue λ of A having the largest real part is unique, with u an eigenvector corresponding to λ, and if the initial condition satisfies $u^*AY_0 \neq 0$, then the solution Y of the initial value problem (3.10) satisfies:*

$$\lim_{t\to\infty} \frac{Y(t)}{\|Y(t)\|_2} = \pm u \tag{3.17}$$

Proof:
Since the exact solution of problem (3.10) is $Y(t) = e^{At}Y_0$, it follows directly from proposition 5 that $\lim_{t\to\infty} \frac{e^{tA}Y_0}{\|e^{tA}Y_0\|_2} = \pm u \Longleftrightarrow \lim_{t\to\infty} \frac{Y(t)}{\|Y(t)\|_2} = \pm u$. ■

3.4.2 Behavior of the Solution and Choice of an EOS Condition

The global behavior of any solution of the linear problem $dY/dt = AY$ depends on the spectrum of A, as described in [52].
The EOS condition is then chosen, according the behavior of the solution:

1. In the case where *some eigenvalues of A have positive real parts, and that having the largest real part is real, with multiplicity 1*, then the global behavior of the solution is explosive, in the direction of the eigenvector corresponding to this eigenvalue. The EOS condition is then chosen similarly to that given in Chapter 2, for explosive problems.

2. In the case where *some eigenvalues of A have positive real parts, and those having the largest real part are two complex conjugate eigenvalues, with multiplicity 1*, then the global behavior of the solution is explosive, in an oscillatory way in the subspace spanned by the corresponding eigenvectors. The EOS condition can then be chosen either similarly to that given in section 2.2, for explosive problems or similarly to that given in section 2.1, for oscillatory problems.

3. In the case where *all eigenvalues of A have negative real parts*, then the global behavior of the solution is extinctive. The EOS condition is chosen similarly to that given in Chapter 2, for extinctive problems.

4. In the case where *all eigenvalues of A have non positive real parts, and if the eigenvalues of which the real part is zero have an algebraic multiplicity equal to 1*, then the solution is bounded. The EOS condition is chosen similarly to that given in Chapter 2, for bounded problems.

5. In the case where *all eigenvalues of A are imaginary numbers*, then the global behavior of the solution is periodic. A slice ends after completion of a full period.

3.4.3 Rescaling & Similarity Properties

Rescaling the problem is done by applying the change of variables (1.5):

$$\begin{cases} t = T_{n-1} + \beta_n s, \\ Y(t) = Y_{n-1} + D_n Z_n(s), \end{cases}$$

where D_n is the diagonal matrix associated with the vector $\alpha_n \in \mathbb{R}^K$ that is defined in (1.6), in terms of the initial value of the solution Y_{n-1} as follows:

$$\alpha_{n,i} = \begin{cases} Y_{n-1,i} & \text{if } Y_{n-1,i} \neq 0 \\ 1 & \text{if } Y_{n-1,i} = 0 \end{cases}$$

Assuming the invariance of the functions $\{H_n\}$ governing the EOS condition, the Initial Value Problem (3.10) is equivalent to the following sequence of rescaled Initial Value Shooting Problems:

$$\begin{cases} \frac{dZ_n}{ds} = A_n \left(D_n^{-1} Y_{n-1} + Z_n \right) = G_n(Z_n), \ 0 < s < s_n, & (3.18.1) \\ Z_n(0) = 0, & (3.18.2) \\ H\left[Z_n(s)\right] \neq 0, \ \forall s < s_n \text{ and } H\left[Z_n(s_n)\right] = 0, & (3.18.3) \end{cases} \quad (3.18)$$

where

$$A_n = \beta_n D_n^{-1} A D_n \quad (3.19)$$

is a matrix in $\mathbb{R}^{K \times K}$, constant on each n^{th} slice, since β_n and D_n depend only on the starting value Y_{n-1} at the n^{th} slice.

Proposition 6 ***: Asymptotic Similarity***
Under the assumptions of corollary 4 and the nonzeroness condition (1.11), if the sequence $\{\beta_n\}$ is convergent, then the rescaled systems (3.18) are asymptotically similar to the limit problem:

$$\begin{cases} \frac{dZ_L}{ds} = G_L(Z_L), \ 0 < s \leq s_L, & (1.25.1) \\ Z_L(0) = 0, & (1.25.2) \\ H\left[Z_L(s)\right] \neq 0, \ \forall s < s_L \ \text{ and } \ H\left[Z_L(s_L)\right] = 0, & (1.25.2) \end{cases} \quad (3.20)$$

where:

$$G_L(Z_L) = \lambda \exp\left(\beta_L s \lambda\right) \mathbf{1},$$

with $\mathbf{1}$ *being the vector of ones in* $\mathbb{R}^K$, λ *being the real eigenvalue of* A *having the largest real part and* $\beta_L = \lim_{n \to \infty} \beta_n$.

Proof:
The exact solution to each of the rescaled problems (3.18) is given by:
$D_n^{-1} Y_{n-1} + Z_n(s) = e^{A_n s} D_n^{-1} Y_{n-1}$, where $A_n = \beta_n D_n^{-1} A D_n$.
It follows for all n and for all $s \in (0, s_n]$:

$$\begin{aligned} Z_n(s) &= \left(e^{D_n^{-1} \beta_n A D_n s} D_n^{-1} Y_{n-1} - D_n^{-1} Y_{n-1} \right) = \left(D_n^{-1} e^{\beta_n A s} D_n - I \right) D_n^{-1} Y_{n-1} \\ &= D_n^{-1} \left(e^{\beta_n A s} - I \right) D_n D_n^{-1} Y_{n-1}, \end{aligned}$$

yielding:

$$Z_n(s) = D_n^{-1} \left(e^{\beta_n} e^{As} - I \right) Y_{n-1}. \quad (3.21)$$

Let $u_{n-1} = \frac{Y_{n-1}}{\|Y_{n-1}\|_\infty}$ and let U_n be the diagonal matrix having u_{n-1} on its main diagonal. Note that the nonzeroness condition on $\{Y_n\}$ implies the nonzeroness of $\{u_n\}$ and makes U_n invertible. Equation (3.21) can then be written as:
$Z_n(s) = D_n^{-1} \|Y_{n-1}\|_\infty \left(e^{\beta_n A s} - I \right) u_{n-1}$, yielding:

$$Z_n(s) = U_n^{-1} \left(e^{\beta_n A s} - I \right) u_{n-1}. \quad (3.22)$$

Under the assumptions of corollary 4, the eigenvalue λ of A having the largest real part is unique. By letting u be an eigenvector of A corresponding to λ and U be the diagonal matrix having u on its main diagonal, it follows from proposition 5 that:

$$\lim_{n\to\infty} u_{n-1} = u, \tag{3.23}$$

and therefore: $\lim_{n\to\infty} U_n = U$.
It follows, by letting β_L be the limit of the convergent sequence $\{\beta_n\}$ of time-rescaling factors, that the exact solution $Z_n(s)$, given in (3.22) approaches the solution:

$$Z_L(s) = U^{-1}\left(e^{\beta_L As} - I\right)u = U^{-1}\left(e^{\beta_L s\lambda} - 1\right)u = \left(e^{\beta_L s\lambda} - 1\right)\mathbf{1}, \tag{3.24}$$

of the limit problem (3.20).
Hence the asymptotic similarity of the rescaled systems, according to definition 3. ■

Proposition 7 ***: Uniform Similarity of the initial value problems*** (IVP_n)
The critical choice of β_n given, for a given $c_0 > 0$, by:

$$\forall n \geq 1, \qquad \beta_n = \frac{c_0}{\left\|D_n^{-1}AD_n\right\|_\infty}, \tag{3.25}$$

results in a sequence $\{\beta_n\}$ that is uniformly upper bounded and makes the initial value problems:

$$(IVP_n) \quad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

uniformly similar, according to definition 5 of Chapter 1.

Proof:
Let $\rho(A)$ be the spectral radius of the matrix A, and therefore that of $D_n^{-1}AD_n$.
Since $\left\|D_n^{-1}AD_n\right\| \geq \rho(D_n^{-1}AD_n) = \rho(A)$, it follows directly:

$$\forall n \geq 1, \quad \beta_n = \frac{c_0}{\left\|D_n^{-1}AD_n\right\|_\infty} \leq \frac{1}{\rho(A)},$$

proving the uniform upper boundedness of $\{\beta_n\}$.

On the other hand, one has: $\forall i, j \in \{1, 2, \cdots, K\}$, $\left(D_n^{-1}AD_n\right)_{ij} = a_{ij}\frac{\alpha_{n,j}}{\alpha_{n,i}}$.
The critical choice (3.25) for $\{\beta_n\}$ expresses then explicitly as:

$$\forall n \geq 1, \qquad \beta_n = \frac{c_0}{\max_{1\leq i\leq K}\sum_{j=1}^{K}\left|a_{ij}\frac{\alpha_{n,j}}{\alpha_{n,i}}\right|}. \tag{3.26}$$

In this linear case, the i^{th} component of $G_n(Z_n) = \beta_n D_n^{-1}AD_n\left(D_n^{-1}Y_{n-1} + Z_n\right)$ is:
$\left(G_n(Z_n)\right)_i = \beta_n\alpha_{n,i}^{-1}\sum_j a_{ij}\alpha_{n,j}\left(\frac{Y_{n-1,j}}{\alpha_{n,j}} + Z_{n,j}\right)$, or equivalently:

$$\left(G_n(Z_n)\right)_i = \sum_{j=1}^{K}\gamma_{n,ij}\left(\delta_{n,j} + Z_{n,j}\right), \tag{3.27}$$

where $\gamma_{n,ij} = \beta_n a_{ij} \frac{\alpha_{n,j}}{\alpha_{n,i}}$ and $\delta_{n,j} = \frac{Y_{n-1,j}}{\alpha_{n,j}} = \begin{cases} 1, & \text{if } Y_{n-1,j} \neq 0 \\ 0, & \text{if } Y_{n-1,j} = 0 \end{cases}$.
It follows a constant jacobian on each slice:

$$[J_{G_n}(Z_n)]_{im} \;=\; \frac{\partial \left(G_n(Z_n)\right)_i}{\partial Z_{n,m}} \;=\; \gamma_{n,im}. \tag{3.28}$$

But the choice (3.26) for $\{\beta_n\}$ results in parameters $\gamma_{n,ij}$ satisfying: $\forall n,\ \forall i,j,\ |\gamma_{n,ij}| \leq c_0$.
It follows then, directly from (3.28), that:

$$\|J_{G_n}(Z_n)\|_\infty \;\leq\; \max_i \sum_{m=1}^{K} \left| \frac{\partial \left(G_n(Z_n)\right)_i}{\partial Z_{n,m}} \right| \;=\; \max_i \sum_{m=1}^{K} |\gamma_{n,im}| \;\leq\; Kc_0 \;= C',$$

and it follows from (3.27) that if $||Z_n(s)||_\infty \leq S\,(\forall s)$, then:

$$\sup_{||Z_n||_\infty \leq S} \{\|G_n(Z_n)\|_\infty\} \leq \max_i \sum_{j=1}^{K} (1+S) \;\leq\; Kc_0(1+S) \;=\; C.$$

Hence the uniform similarity of the initial value problems (IVP_n). ∎

Corollary 5 *Under the assumptions of corollary 4 and the nonzeroness condition (1.11):*

$$\forall n, \forall i : 1 \leq i \leq K,\, Y_{n,i} \neq 0,$$

the critical choice (3.25) yields a convergent sequence $\{\beta_n\}$ and makes the rescaled systems (3.18) asymptotically similar to the limit problem (3.20).

Proof:
The critical choice (3.25) for β_n can be expressed, in case of nonzeroness, as:
$\forall n \geq 1, \qquad \beta_n = \dfrac{c_0}{\max_i \sum_{j=1}^{K} \left| a_{ij} \frac{Y_{n-1,j}}{Y_{n-1,i}} \right|} = \dfrac{c_0}{\max_i \sum_{j=1}^{K} \left| a_{ij} \frac{\frac{Y_{n-1,j}}{\|Y_{n-1}\|_\infty}}{\frac{Y_{n-1,i}}{\|Y_{n-1}\|_\infty}} \right|} = \dfrac{c_0}{\max_i \sum_{j=1}^{K} \left| a_{ij} \frac{u_{n-1,j}}{u_{n-1,i}} \right|}.$
Since $\lim_{n\to\infty} u_{n-1} = \pm u$, it follows that:

$$\lim_{n\to\infty} \beta_n = \beta_L, \qquad \text{with} \qquad \beta_L = \frac{1}{\max_{i,j} \left\{ |a_{ij}| \left| \frac{u_j}{u_i} \right| \right\}}. \tag{3.29}$$

Proposition 6 yields then, directly, the asymptotic similarity of the rescaled systems.

∎

The following theorem is in line with theorem 3 of Chapter 2 and its corollary.

Theorem 5 ***: Uniform Similarity of*** (S'_n) ***- Particular explosive linear case***
If:

1. *the matrix A is tridiagonal:*

$$A = \begin{pmatrix} d_1 & u_1 & 0 & \cdots & 0 \\ l_2 & d_2 & u_2 & \ddots & \vdots \\ 0 & l_3 & d_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & u_{K-1} \\ 0 & \cdots & 0 & l_k & d_k \end{pmatrix}$$

and satisfies:
$$\begin{cases} \forall i \in 1, 2, \cdots, K,\ d_i > 0, \\ \forall i \in 1, 2, \cdots, K-1,\ u_i < 0, \\ \forall i \in 2, 3, \cdots, K,\ l_i < 0, \\ \forall i \in 1, 2, \cdots, K,\ d_i > |l_i| + |u_i| \qquad (\textit{letting } l_1 = 0 \textit{ and } u_K = 0) \end{cases},$$

2. *$||Y(t)||_\infty$ monotonously increases with t and $\forall i,\ \forall t,\ Y_i(t) > 0$,*

3. $\exists i_0 \in \{1, 2, \cdots, K\}$, $\begin{cases} \forall i,\ \forall t, \quad 0 < Y_i(t) \leq Y_{i_0}(t), \\ \lim_{t\to\infty} Y_{i_0}(t) = \infty, \\ \lim_{t\to\infty} F_{i_0}(Y(t)) = \infty, \\ F_{i_0}(Y(t)) \textit{ and } Y_{i_0}^{-1} F_{i_0}(Y(t)) \textit{ are positive and increasing,} \end{cases}$

4. *the solution $Y(t)$ satisfies the curvature condition:*

$$\forall i \in 1, 2, \cdots, K,\ \forall t > 0, \qquad Y_{i-1}(t) - 2Y_i(t) + Y_{i+1} < 0, \tag{3.30}$$

with $\forall t > 0,\ Y_0(t) = Y_{K+1}(t) = 0$,

5. *the EOS condition is that, specific to explosive problems in the nonzeroness case, given in (2.7) in Chapter 2,*

6. *the sequence $\{\beta_n\}$ is selected, according to (3.25): $\forall n \geq 1,\ \beta_n = \frac{c_0}{\|D_n^{-1} A D_n\|_\infty}$, with:*

$$c_0 = \frac{[d_{i_0} - |l_{i_0}| - |u_{i_0}|]}{3 \max_i \{|l_i|, d_i, |u_i|\}} \tag{3.31}$$

then, the rescaled problems:

$$(S'_n) \quad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \qquad 0 < s \leq s_n, \\ Z_n(0) = 0, \\ \|Z_n(s)\|_\infty < S,\ \forall s < s_n \ \textit{ and } \ \|Z_n(s_n)\|_\infty = S, \end{cases}$$

are uniformly similar, according to definition 6, given in Chapter 1.

Proof:
Note first that the choice (2.12), given in Chapter 2 for $\{\beta_n\}$ and for a given $c_0 > 0$, coincide exactly with the present choice (3.25):

$$\forall n \geq 1, \quad \beta_n = \frac{c_0 Y_{n-1,i_0}}{F_{i_0}(Y_{n-1})} = \frac{c_0}{\left\|D_n^{-1} A Y_{n-1}\right\|_\infty} = \frac{c_0}{\left\|D_n^{-1} A D_n\right\|_\infty}.$$

As stated in proposition 7, the choice (3.25) for $\{\beta_n\}$ yields initial value problems:

$$(IVP_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \\ Z_n(0) = 0, \end{cases}$$

that are uniformly similar.
Together with the EOS condition (2.7), this uniform similarity of (IVP_n) yields a sequence $\{s_n\}$ of rescaled time values at the end of slices that is uniformly lower bounded, according to proposition 1 of Chapter 2: $\exists \underline{s} > 0, \ \forall n \geq 1, \ \underline{s} \leq s_n$.

It suffices now to prove the uniform upper boundedness of $\{s_n\}$ for deducing the uniform similarity of the rescaled problems (S'_n).
Under our assumptions, one has for all n:

$$\begin{aligned} \left\| D_n^{-1} A D_n \right\|_\infty &= \max_i \left\{ |l_i| \, \tfrac{Y_{n-1,i-1}}{Y_{n-1,i}} + d_i + |u_i| \, \tfrac{Y_{n-1,i+1}}{Y_{n-1,i}} \right\} \\ &\leq \max_i \left\{ |l_i|, d_i, |u_i| \right\} \left[1 + \tfrac{Y_{n-1,i-1}}{Y_{n-1,i}} + \tfrac{Y_{n-1,i+1}}{Y_{n-1,i}} \right] \end{aligned}$$

Letting $c = \max_i \{|l_i|, d_i, |u_i|\}$, it follows:

$$\left\| D_n^{-1} A D_n \right\|_\infty \leq c \left[3 + \frac{Y_{n-1,i-1} - 2Y_{n-1,i} + Y_{n-1,i+1}}{Y_{n-1,i}} \right].$$

Using now the curvature condition (3.30), one gets: $\left\| D_n^{-1} A D_n \right\|_\infty \leq 3c$ and deduces:

$$\forall n, \quad \beta_n \geq \frac{1}{3c}. \tag{3.32}$$

Besides, our assumptions give for all $i \in 1, 2, \cdots, K$:

$$(G_n(Z_n))_i = \beta_n Y_{n-1,i}^{-1} \left[l_i Y_{n-1,i-1} \left(1 + Z_{n_i - 1} \right) + d_i Y_{n-1,i} \left(1 + Z_{n,i} \right) + u_i Y_{n-1,i+1} \left(1 + Z_{n,i+1} \right) \right]$$

Applying this to the component i_0 and at $s = 0$, at which $Z_n(s) = 0$ $(\forall n)$, yields:

$$(G_n(0))_{i_0} = \beta_n Y_{n-1,i_0}^{-1} \left[l_{i_0} Y_{n-1,i_0 - 1} + d_{i_0} Y_{n-1,i_0} + u_{i_0} Y_{n-1,i_0 + 1} \right]$$

$$(G_n(0))_{i_0} = \beta_n \left[d_{i_0} + l_{i_0} \frac{Y_{n-1,i_0-1}}{Y_{n-1,i_0}} + u_{i_0} \frac{Y_{n-1,i_0+1}}{Y_{n-1,i_0}} \right]$$

But assumption 1 of the theorem makes: $\frac{Y_{n-1,i_0-1}}{Y_{n-1,i_0}} \leq 1$, $\frac{Y_{n-1,i_0+1}}{Y_{n-1,i_0}} \leq 1$, $l_{i_0} = -|l_{i_0}|$ and $u_{i_0} = -|u_{i_0}|$. Together with (3.32), it leads to:

$$(G_n(0))_{i_0} \geq \frac{1}{3c} \left[d_{i_0} - |l_{i_0}| - |u_{i_0}| \right] = c_0 > 0.$$

As done in theorem 3 of Chapter 2, and because $\frac{dZ_{n,i_0}}{ds} = G_{n,i_0}[Z_n(s)]$, one deduces:

$$\forall n, \ \ \forall s > 0 \qquad Z_{n,i_0}(s) = \int_0^s G_{n,i_0}[Z_n(\sigma)] \, d\sigma \geq c_0 s,$$

yielding at $s = s_n$, where $\forall n, \ Z_{n,i_0}(s_n) = S$ (since the EOS condition is reached at the component i_0, $\forall n$, due to assumption 2 of the theorem): $\forall n \geq 1, \quad s_n \leq \frac{S}{c_0} = \bar{s}$. ∎

Part II

RaPTI Algorithm & Applications

Chapter 4

Ratio Property & RaPTI Algorithm

The goal of this chapter is to devise a parallel algorithm for solving some initial value problems of the form:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = F(Y), & t > 0, \\ Y(0) = Y_0. \end{cases}$$

As described in Chapter 1, our approach for time-slicing is based on obtaining a coarse grid of time slices $\{[T_{n-1}, T_n]\,,\ n \geq 1\}$ where each of the slices is defined by an end-of-slice condition. Then, the rescaling methodology changes, on every n^{th} time-slice, the variables $(t, Y(t))$ into $\{(s, Z_n(s))\}$, using $t = T_{n-1} + \beta_n s_n$ and $Y(t) = Y_{n-1} + D_n Z_n(s_n)$. This makes (S) equivalent to a *sequence* $\{(S'_n)\}_{n\geq 1}$ of *rescaled* initial value *shooting* problems:

$$(S'_n) \qquad \begin{cases} \frac{dZ_n}{ds} = G_n(Z_n) & = \beta_n D_n^{-1} F\big(Y_{n-1} + D_n Z_n\big), & 0 < s \leq s_n \\ Z_n(0) = 0, & & \\ H\left[Z_n(s)\right] \neq 0, & \forall s < s_n & \text{and} \quad H\left[Z_n(s_n)\right] = 0, \end{cases}$$

where the EOS condition is assumed to be governed by an invariant function H.

Those rescaled systems can be solved independently and in parallel, provided one can get predictions $\{Y_{n-1}\}$ for the starting values of the solution, on each time-slice.

Enabling predictions that use the rescaling methodology and its similarity properties is at the heart of our new approach for a parallel-in-time algorithm. This approach is further strengthened on the basis that the rescaling technique allows to deal in parallel with the sequence of rescaled initial value shooting problems (S'_n), with the local rescaled time $s \in [0, s_n]$. Likewise the sequence $\{s_n\}$, which is a priori unknown and results from reaching the EOS condition, the sequence $\{T_n\}$ is also unknown and finalized, after $\{s_n\}$, using $T_0 = 0$ and $T_n = T_{n-1} + \beta_n s_n$.

Under the nonzeroness condition (1.11), that assumes: $\forall n, \forall i : 1 \leq i \leq K, Y_{n,i} \neq 0$, predictions of $\{Y_n\}$ start with the end-of-slice relation:

$$Y_n = D_n \left(\mathbf{1} + Z_n(s_n)\right).$$

Letting:

$$R_n = \mathbf{1} + Z_n(s_n),$$

makes the previous relation equivalent to:

$$Y_n = D_n R_n,$$

which states that Y_n results from a simple components multiplication of the vectors R_n and Y_{n-1} (Hadamard product):

$$\forall i, \quad Y_{n,i} = R_{n,i} Y_{n-1,i} \qquad \Longleftrightarrow \qquad Y_n = D_{R_n} Y_{n-1},$$

where D_{R_n} is the diagonal matrix having R_n on its main diagonal.
This shows that R_n can be viewed as a "ratio-vector" between the consecutive starting values Y_{n-1} and Y_n of the solution.

Besides, the similarity properties, given in definitions (2), (3) and (4) of chapter 1, yield a *near*-invariance of the rescaled end-of-slice values $\{Z_n(s_n)\}$ and consequently of the ratio-vectors $\{R_n\}$. This makes the starting values of the solution $\{Y(T_n)\} = \{Y_n\}$, at the beginning of each time-slice, well-suited to be predicted!

This rationale leads to a "Ratio-based Parallel Time Integration" procedure which main features are as follows:

1. A *preliminary sequential analysis*, that is manually done on the rescaled systems resulting from an initial value problem (S) of which *the solution is known to have a given behavior*, and on which rescaling yields one of the similarity properties given in definitions (2), (3) or (4) of chapter 1. It consists in:

   (a) A sequential run that solves a certain number of rescaled systems (S'_n), for $n = 1, 2, \cdots, n_s$. The slice number n_s results from a "ratios stabilization test" expressing the detection of a ratio property.

   (b) Based on the obtained ratios, one builds a mathematical model fitting $\{Z_n(s_n)\}$, or equivalently $\{R_n\}$, as a function of n.

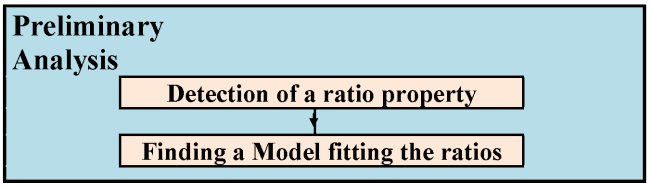


Figure 4.1: Preliminary Analysis, preceding RaPTI

2. A *predictor-corrector scheme* (RaPTI algorithm) that can then be implemented on parallel architectures and consists in:

   (a) Run sequentially, until reaching the ratio property, at slice number n_s.

   (b) Get the mathematical model's parameters with the obtained ratios, and use it for predicting the starting values $\{Y_n\}$, for $n > n_s$.

   (c) Solve, *in parallel*, the rescaled systems (S'_n), for $n > n_s$, with functions $\{G_n\}$ based on the predicted values of $\{Y_n\}$. This is done as long as the corrected end-of-slice values $\{Z_n(s_n)\}$ match with the predicted ones.

   (d) Update n_s and iterate steps (b) and (c) until the total number N of slices is solved.

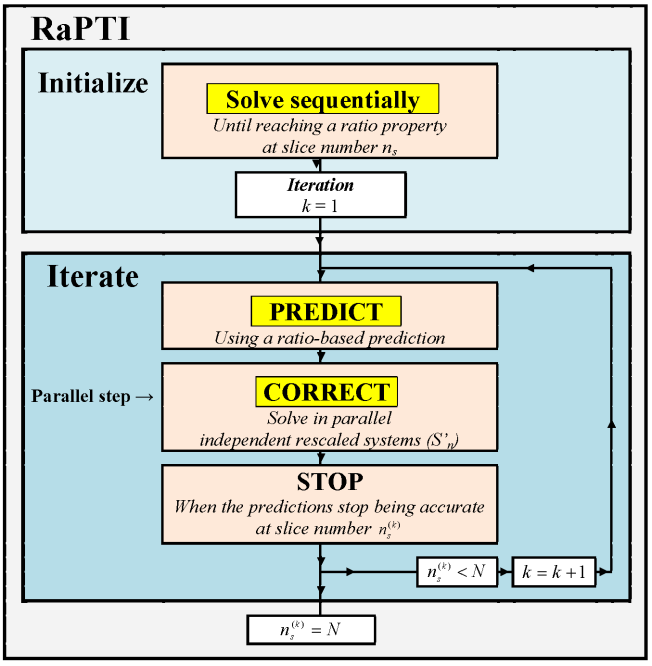


Figure 4.2: RaPTI Algorithm - A New Parallel-in-Time Scheme

As shown in the numerical results given in chapters 5, 6 and 7, RaPTI algorithm results, whenever it applies, in a speed-up that is close to the maximum speed-up stated by Amdahl's law. This is mainly due to the following facts:
(i) The ratio-based prediction procedure yields predicted values that are close to the exact ones, especially in the case of asymptotic similarity, attaining therefore the convergence of the scheme after a small number of iterations.
(ii) The operations involved in the ratio-based prediction procedure are not costly, consisting, in general, of a polynomial fit of a few ratio-vectors that is used for extrapolating the following ratios.
(iii) Parallel computations are stopped, within an iteration, as soon as one of the slices fails the matching test, reducing the overhead resulting from needlessly corrected slices (unlike other parallel methods).

This chapter is divided as follows.

In *section 1*, we show how the rescaling methodology, with the various types of similarity properties defined on rescaled systems, in chapter 1, could yield corresponding types of "*ratio properties*".

In *section 2*, we show how the invariance property of the rescaled systems results in a rare perfect parallel algorithm. In such a case, RaPTI can be applied for benchmarking, only. A specific sequential algorithm (RaTI), using the invariance property, is proposed for solving the rescaled systems.
All other sections deal with non-invariant cases.

In *section 3*, the sequential analysis that should be done, preliminary to the application of RaPTI, is discussed. It aims at justifying the applicability of the method and providing a model fitting the ratios as a function of n.

In *section 4*, we show how the rescaled systems are solved in parallel, using predicted functions $\{G_n\}$ and getting corrected end-of-slice values. A mathematical analysis intends then to provide a "convergence test" that should hold in order to guarantee the convergence of the corrected solution toward the exact one.

Section 5 starts with an overview of RaPTI algorithm, together with some parallel implementation techniques, allowing a speed-up improvement. Then, all the modules of the algorithm are detailed.

Finally, in *section 6*, some choices concerning the numerical integration are given.

Appendix 1 details the implementation of the procedures mentioned in this chapter.
Appendix 2 provides a small analysis of the communication cost of RaPTI algorithm, as well as the way for evaluating the speed-up and efficiency. It gives also a quick description of the parallel architecture and the programming language that have been used in the numerical experiments.

4.1 Similarity and Ratio Property

4.1.1 Ratio Vectors and End-of-slice Relations

Recall that for all n, $Y_n = Y(T_n)$ and $\alpha_n \in \mathbb{R}^K$ is defined in (1.6) as:

$$\alpha_{n,i} = \begin{cases} Y_{n-1,i} & \text{if } Y_{n-1,i} \neq 0 \\ 1 & \text{if } Y_{n-1,i} = 0 \end{cases}$$

D_n is then the diagonal invertible matrix having the vector $\alpha_n \in \mathbb{R}^K$ on its main diagonal:

$$D_n = \text{diag}(\alpha_n) \in \mathbb{R}^{K \times K} \tag{4.1}$$

In case the nonzeroness condition (1.11) is satisfied:

$$\forall n, \forall i : 1 \leq i \leq K,\, Y_{n,i} \neq 0,$$

then $\alpha_n = Y_{n-1}$ $(\forall n)$ and D_n is simply the diagonal matrix having Y_{n-1} on its main diagonal.

Definition 8 ***: Ratio-Vectors***
Under the nonzeroness condition (1.11), the ratio-vector, on the n^{th} slice, is defined by:

$$R_n = D_n^{-1} Y_n, \tag{4.2}$$

that is the ratio of the end-of-slice value to the starting value of the solution vector, component wise given as:

$$\forall i \in \{1, 2, .., K\}, \quad R_{n,i} = \frac{Y_{n,i}}{Y_{n-1,i}}, \tag{4.3}$$

Resulting Recurrence:
Under the nonzeroness condition, one has the recurrence relation:

$$\forall n \geq 1, \quad \forall i \in \{1, 2, .., K\}, \quad Y_{n,i} = R_{n,i} Y_{n-1,i}, \tag{4.4}$$

yielding:

$$\forall n \geq 1, \quad \forall i \in \{1, 2, .., K\}, \quad Y_{n,i} = R_{n,i} R_{n-1,i} \cdots R_{1,i} Y_{0,i}. \tag{4.5}$$

The vector form of this recurrence is:

$$Y_n = D_{R_n} Y_{n-1}, \tag{4.6}$$

where D_{R_n} is, for all n, the diagonal matrix having R_n on its main diagonal, yielding:

$$Y_n = D_{R_n} D_{R_{n-1}} \cdots D_{R_1} Y_0. \tag{4.7}$$

Rescaled Ratio Vectors:
The change of variable (1.5.2) yields:

$$\text{at } t = T_n, \quad Y_n = Y_{n-1} + D_n Z_n(s_n).$$

Under the nonzeroness condition, this relation becomes:

$$\forall n, \; Y_n = D_n \left[\mathbf{1} + Z_n(s_n) \right], \tag{4.8}$$

where $\mathbf{1} \in \mathbb{R}^K$ is a vector of ones, and therefore:

$$Z_n(s_n) = D_n^{-1} Y_n - \mathbf{1} = R_n - \mathbf{1},$$

or equivalently:

$$\forall n,\ \forall \beta_n, \quad R_n = \mathbf{1} + Z_n(s_n). \tag{4.9}$$

One deduces that $D_{R_n} = D_{\mathbf{1}+Z_n(s_n)} = I + D_{Z_n(s_n)}$, where I is the identity matrix and $D_{Z_n(s_n)}$ is the diagonal matrix associated with the vector $Z_n(s_n)$.
Then, the recurrence (4.6) rescales to:

$$\forall n,\ Y_n = \left(I + D_{Z_n(s_n)}\right) Y_{n-1}. \tag{4.10}$$

It follows:

$$Y_n = \left(I + D_{Z_n(s_n)}\right) \cdots \left(I + D_{Z_1(s_1)}\right) Y_0. \tag{4.11}$$

4.1.2 Ratio Properties Deriving from Similarity Properties

Proposition 8 ***: Similarity and Ratio Properties***
Under the nonzeroness condition (1.11), one has the following results:

1. *Invariant rescaled systems, according to definition 2 of chapter 1, yield a sequence $\{Y_n\}$ of end-of-slice values of the solution having a* ***perfect ratio property*** *in the sense that the sequence $\{R_n\}$ of ratio-vectors is constant:*

$$\forall n > 0, \quad R_n \ = \ R_1. \tag{4.12}$$

2. *If the rescaled systems (S'_n) have the property of asymptotic similarity, according to definition 3 of chapter 1, and if the end-of-slice values of the rescaled solution satisfy:*

$$\lim_{n\to\infty} Z_n(s_n) = Z_L(s_L), \tag{4.13}$$

then the sequence $\{Y_n\}$ of end-of-slice values of the solution has an ***asymptotic ratio property****, in the sense that the sequence $\{R_n\}$ of ratio-vectors is convergent:*

$$\lim_{n\to\infty} R_n = R_L \in \mathbb{R}^K. \tag{4.14}$$

3. *Rescaled systems presenting a weak similarity on n_r consecutive slices, up to a tolerance ϵ, according to definition 4 of chapter 1, yield, on those slices, a sequence $\{Y_n\}$ of end-of-slice values of the solution satisfying a* ***weak ratio property****, up to the same tolerance ϵ, in the sense that:*

$$\forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \quad \|R_n - R_{n-1}\|_\infty < \epsilon. \tag{4.15}$$

proof:

1. The proof follows directly from the definition of invariance stating that $\forall n > 0,\ G_n(.) = G_1(.)$, making all the rescaled problems equivalent to the unique initial value shooting problem (S'_1). Therefore: $\forall n > 0, \quad Z_n(s_n) = Z_1(s_1)$, where $Z_1(s_1)$ is the end-of-slice value of the rescaled solution of (S'_1).
   Since $\forall n > 0,\ R_n = \mathbf{1} + Z_n(s_n)$, it follows that: $\forall n > 0,\ R_n \ = \ R_1$.

2. The asymptotic similarity, together with (4.13), yields:

$$\lim_{n\to\infty} R_n = \lim_{n\to\infty} [\mathbf{1} + Z_n(s_n)] = \mathbf{1} + \lim_{n\to\infty} Z_n(s_n) = \mathbf{1} + Z_L(s_L) = R_L.$$

3. The proof follows directly from the definition of weak similarity on n_r consecutive slices and from relation (4.9).

■

Note that ratio properties are induced by a relevant choice of an EOS condition, that completely determines the end-of-slice values of the solution $\{Y_n\}$ and are not related to any rescaling. However, the above proposition relates the similarity properties of the rescaled systems to the ratio properties. This shows how the time-slicing and rescaling techniques are complementary: (i) the first one providing a sequence $\{(S_n)\}$ of slice-based problems where the starting values $\{Y_n\}$ may have a ratio property, making them predictable; (ii) the second one allowing to use the predictions for solving the rescaled systems $\{S'_n\}$, in parallel, using a local time s that starts at 0.

Remark 5
The ideal perfect ratio property is usually met in academic problems only (and is not of big interest in applications). Asymptotic ratio property can be met in some application problems such as those considered in Chapters 5 and 6.
Both are stronger (particular) cases of the weak ratio property that is more likely to be met and is essential for a ratio-based prediction procedure.
An application problem yielding a weak ratio property is considered in Chapter 7.

Remark 6
It is important to notice that uniform similarity, which is of great numerical interest for solving stiff systems, is not enough for yielding a ratio property.

4.2 Case of Invariance

In the case of invariant rescaled systems, yielding a **perfect ratio property**, i.e. a constant sequence $\{R_n\}$ of ratio-vectors ($\forall n > 0,\ R_n = R_1$), the sequential computation of the solution on only *one slice* allows to get the **exact** starting values $\{Y_n\}$ of the solution on all slices, using the recurrence (4.7):

$$\forall n, \quad Y_n = (D_{R_1})^n Y_0 = \left(I + D_{Z_1(s_1)}\right)^n Y_0.$$

Hence, implementing RaPTI algorithm on invariant cases yields a perfect parallelism with no need of any iteration or communication!

However, the parallelism is of no need here and applying RaPTI algorithm serves for benchmarking only. This is due to the fact that solving the problem on the first slice solves it on all slices, as described in the following Ratio-based Time Integration (RaTI) algorithm, for invariant cases.

RaTI algorithm (Sequential - For invariant cases)

- **Step 1:** **Solve the First Slice**

  getting the rescaled time s_1 and the rescaled solution $Z_1 : [0, s_1] \longrightarrow \mathbb{R}^k$, and deduce T_1 and $Y_1 : [0, T_1] \longrightarrow \mathbb{R}^K$.

- **Step 2:** **Get the solution on all slices**

  Since the rescaled solution $Z_1 : [0, s_1] \longrightarrow \mathbb{R}^k$ of the first time-slice solves all the slices, then the solution on all time-slices is obtained by simply applying recurrently, onto the rescaled solution of problem (S_1') and for all $n > 1$, the change of variables:

$$\begin{cases} t = T_{n-1} + \beta_n s, & \beta_n > 0 \\ Y(t) = D_n \left(\mathbf{1} + Z_1(s)\right), \end{cases}$$

In all remaining sections of this chapter, the very particular case of invariance is no longer considered.

4.3 Preliminary Sequential Analysis

Consider an initial value problem (S) of which the solution is known to have a given behavior and on which rescaling yields rescaled systems having one of the similarity properties given in definitions (2), (3) or (4) of chapter 1.

Prior to any application of our ratio-based parallel scheme (RaPTI Algorithm), we carry a preliminary sequential analysis that has three goals:

1. Justify the *applicability* of RaPTI, by detecting a ratio property: this is done by solving sequentially the rescaled initial value shooting problems (S'_n) until the end-of-slice values $\{Z_n(s_n)\}$ stabilize at slice number n_s, signifying that the ratios $\{R_n\}$ stabilize, i.e. a ratio property has been reached.

2. Extract from $\{R_n\}_{n\leq n_s}$ a *model* that is likely to give an accurate estimation for $\{R_n\}_{n>n_s}$.

3. Get a first estimate of the *total number N of slices* that should be solved for covering a given interval $[0, T]$ of integration.

After these sequential computation and analysis on n_s slices, one can then estimate the starting values $\{Y_n\}_{n>n_s}$ of the solution and envisage solving (S'_n), in parallel, on the following slices $\{n_s + 1, n_s + 2, \cdots, N\}$.

It should be noticed, as indicated by all our numerical experiments, that the model found in this preliminary analysis is usually appropriate to be used after each iteration of RaPTI, with updates based on the new values of n_s.

4.3.1 Reaching a Ratio Property

1. **Weak Similarity: Numerical Detection of a Ratio Property**
   When solving sequentially the rescaled systems (S'_n), in case the sequence $\{R_n\}$ is not convergent, one can sometimes guess a weak ratio property.

   Let n_r be a predefined number of slices. A ratio property, up to a tolerance ϵ, is said to be detected on n_r consecutive slices and reached at slice number $n_s = n_0 + n_r$, if the relation (4.15) is satisfied:

   $$\forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \quad \|R_n - R_{n-1}\|_\infty < \epsilon.$$

   Note that this numerical verification of a ratio property does not mean that the property will hold for all $n > n_s$. However, a relevant choice of the uniform EOS condition that suits the global behavior of the physical problem, together with the similarity of the rescaled systems, are assumed to make the ratio property hold beyond n_s.

   The search for a ratio property, up to a tolerance ϵ, is done by means of a procedure called "DETECT_RATIO_PROPERTY_UP_TO_ϵ", detailed in appendix 1, that stops if the property is not reached at a maximum acceptable slice number n_{stop}.

Note that, in case a ratio property is reached, advancing the sequential computations on n_s slices should provide enough data for the analysis of ratios evolution, intended to find a mathematical model fitting the ratios.

2. **Asymptotic Similarity: Reaching a Ratio Property**
In case of asymptotic similarity, the ratio-vectors $\{R_n\}$ are known, under the nonzeroness condition (1.11), to converge toward a limit ratio R_L and the ratio property is mathematically proved. This limit ratio can be calculated by solving the limit problem (consisting of a unique time-slice).

Since asymptotic similarity guarantees a weak similarity, one can choose to reach a numerical ratio property, using the procedure described above:
DETECT_RATIO_PROPERTY_UP_TO_ϵ.
The ratio property is here guaranteed to be detected.

An alternative (and more severe) choice would be to reach the limit ratio R_L, up to ϵ, on n_r consecutive slices:

$$\forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \quad \|R_n - R_L\|_\infty < \epsilon.$$

The previous procedure translates then to another one, given in appendix 1, called: "REACH_ASYMPTOTIC_RATIO_PROPERTY_UP_TO_ϵ".

4.3.2 Ratios Predictive Model

The present section deals with both cases of asymptotic and weak similarity and includes an additional improvement specific to the case of asymptotic similarity.

The basic idea of a ratio-based prediction procedure is to assume that the existence of a ratio property makes the sequence of ratios follow a *pattern* that could be expressed by a mathematical model.
For that purpose and after the preliminary sequential computations on n_s slices (allowing to reach a ratio property), a data analysis is done on the sequence $\{R_1, \cdots, R_{n_s}\}$ of available exact ratios and several mathematical models for fitting $\{R_n\}$ are devised and compared. Each model j will give a different expression of the n^{th} ratio-vector: $R_{fit_j}(n) \approx R_n$. The models are then evaluated with respect to their ability to predict future ratios: the model extrapolating best onto next slices will be chosen.

- **Simplest Model:**
Since the rescaled systems present a ratio property, that can be understood as a near-invariance of the ratios, then the last computed ratio R_{n_s} can approximate the following ratios that need to be predicted:

$$\forall n \geq n_s, \qquad R^p_n = R_{n_s}. \tag{4.16}$$

This makes the sequence $\{R^p_n\}_{n>n_s}$ constant (as if it was an invariant case) and approximates each component $\{Y^p_{n,i}\}_{n\geq n_s}$ with a geometric sequence:

$$\forall n \geq n_s, \quad \forall i \in \{1, 2, \cdots, k\}, \qquad Y^p_{n,i} = (R_{n_s,i})^{n-n_s}\, Y^e_{n_s,i}. \tag{4.17}$$

where $Y^e_{n_s}$ is the exact sequentially computed end-of-slice value of the solution at $t = T_{n_s}$.

However, even if successive ratios are very close (due to the ratio property), the procedure could yield predicted values that are not good enough.

- **Improved Models:**
  Actually, the slow evolution of the ratios can be anticipated using mathematical models fitting the sequence $\{R_n\}$ $(n \leq n_s)$ and extrapolating well for $n > n_s$. Many mathematical models should be tested and compared.
  Each of them necessitates the following steps.

  - *Choice of a mathematical form:*
    This form could be a explicit function of the n, where n is the number of slice.
    Some examples of explicit forms, tested in this thesis, are:
    * polynomial form of degree d:
    $R_{fit}(n) = a_d n^d + \cdots + a_1 n + a_0$, where $a_0, \cdots, a_d$ are parameters;
    * exponential form: $R_{fit}(n) = be^{an}$, where a, b are parameters;
    * logarithmic form: $R_{fit}(n) = a \log n + b$, where a, b are parameters;
    * power form: $R_{fit}(n) = bn^a$, where a, b are parameters.
    The parameters are determined in a way to fit best the input data in the *least-squares sense*.
    The mathematical model could also consist in a recurrence relation.
    A simple example would be a linear recurrence relation of order d:
    $R_{fit}(n) = a_0 + a_1 R_{fit}(n-1) + \cdots + a_d R_{fit}(n-d)$, where $a_0, \cdots, a_d$ are parameters, determined by solving linear systems obtained by applying the desired recurrence onto the last available ratios.
    However, our numerical experiments have shown that explicit models approximate the ratios better than recurrence relations.
    Specifically, the models that have been chosen, in this thesis, consist in polynomials of degree 1, 2 or 3.
  - *Choice of the number of input data:*
    This number should mainly be enough to determine the parameters of the fit.
    As an example, a polynomial of degree d needs at least $n_{min} = d+1$ input data.
    Thus, $n_{prec} \geq n_{min}$.
    However, and because of the predictive goal of the fit, it seems that, for a given mathematical form, the parameters evaluation yielding the best extrapolation uses only the minimum number n_{min} of last computed ratios, namely those preceding directly the ratios to be predicted.

Case of Asymptotic Ratio Property: Additional Improvement
In case of asymptotic similarity, the ratio-vectors $\{R_n\}$ converge to a limit ratio R_L, as stated in proposition 8. Even though, and because the ratios might converge very slowly toward R_L, the simple model (4.16) could require the sequential run on a very large number n_s of slices for the predicted values to be good enough and, here also, an improved mathematical model could be very useful.
Moreover, in this case of asymptotic similarity, the mathematical model could take into consideration the convergence of $\{R_n\}$ toward R_L by forcing the model to have a limit equal to the limit ratio R_L.
A way for doing this is, for example, to fit $[n\,(R_n - R_L)]$ as a polynomial in the variable $\frac{1}{n}$, of degree d. The result would yield:
$n\,(R_n - R_L) \approx n\,(R_{fit}(n) - R_L) = a_d \left(\frac{1}{n}\right)^d + \cdots + a_1 \left(\frac{1}{n}\right) + a_0,$

and therefore:

$$R_n \approx R_{fit}(n) = R_L + \frac{1}{n}\left[a_d\left(\frac{1}{n}\right)^d + \cdots + a_1\left(\frac{1}{n}\right) + a_0\right], \tag{4.18}$$

showing clearly that:

$$\lim_{n\to\infty} R_{fit}(n) = R_L.$$

Note that a polynomial fit of degree d (in $1/n$) for $[n\,(R_n - R_L)]$, translates to a polynomial fit of degree $(d+1)$ for R_n.
Because of the available limit ratio for showing the road, predictions obtained in the case of asymptotic ratio property are, in practice, much better than those obtained in the case of weak ratio property.

- **Choice of a model:**
  After having been sequentially solved on n_s slices, the problem is again sequentially solved on n_{add} additional time-slices, providing a validation set $\{R_{n_s+1}, \cdots, R_{n_s+n_{add}}\}$ of n_{add} additional exact ratio-vectors. Each j^{th} model is extrapolated over the same n_{add} time-slices, yielding its own approximations of the ratios and comparing them to the validation set thus, getting its maximal error:

$$error_j = \max_{n_s<n\leq n_s+n_{add}}\left[\max_{i\in\{1,2,\cdots,K\}}\left|R_{fit_j,i}(n) - R_{n,i}\right|\right].$$

  The mathematical model yielding the least error is chosen. This is done through the procedure " FIND_RATIOS_MODEL ", given in Appendix 1.

  However, this extrapolation test cannot assess the scope of the model, i.e. it cannot determine in advance the number of slices on which the model extrapolates well. This scope will be the resulting number of converging slices at the first iteration!

It should be noticed that, if necessary, different mathematical models could be chosen for fitting the different components of the ratio-vectors.

4.3.3 Estimation of N

Let $[0, T]$ be the time interval on which the parallel integration of the problem is required. It is important to bear in mind that the size of the time slices to be solved in parallel, using RaPTI algorithm, is a priori unknown and obtained as convergence progresses.
Therefore, once a ratio property has been reached and a predictive mathematical model for the ratios has been devised, one should estimate the total number of slices that is likely to cover the given interval of integration $[0, T]$. This estimation bounds the number of slices to be solved, in case one of the processor does not get a divergent slice, and is updated at each iteration.
Besides, when one deals with explosive problems, as in chapters 5 and 6, an additional condition should stop the solving process as soon as the solution exceeds the machine capacity, even if the given time T is not reached.

Usually, the estimation of N is done using one of the following ways:

- **Refined Estimation of N:**
  This estimation is done while advancing the predictions.

Recall first that, after rescaling, the size of each n^{th} time-slice is given by (1.8) $\forall n,\ \forall \beta_n, \quad \Delta T_n = T_n - T_{n-1} = \beta_n s_n$, with $T_0 = 0$.
This makes the total time of integration covered by N slices equal to $[0, \tilde{T}]$ where:

$$\tilde{T} = \sum_{n=1}^{N} \beta_n s_n. \tag{4.19}$$

At the beginning of each k^{th} iteration, one has the values $\{\beta_n\}_{n \leq n_s^{(k)}}$ and $\{s_n\}_{n \leq n_s^{(k)}}$ only, and should approximate their next values.
Because of the similarity of the rescaled systems, we use $\tilde{s_n}$ as an estimate for s_n, for $n > n_s^{(k)}$, where $\tilde{s_n}$ could be:
(i) $\tilde{s_n} = s_{n_s^{(k)}}$, in general,
(ii) $\tilde{s_n} = \bar{s}$ in case of uniform similarity (upper bound of s_n),
(iii) $\tilde{s_n} = \max\left\{s_{n_s^{(k)}}, s_L\right\}$, in case of asymptotic similarity where s_L is the limit of the convergent sequence $\{s_n\}$.
Besides, since each n^{th} time rescaling factor β_n is expressed in terms of the starting value of the solution Y_{n-1}, it would be easy to predict the values $\{\beta_n\}$, for all $n > n_s^{(k)}$, using the predicted starting values of the solution.
This results in a simultaneous task to be done before each iteration, consisting in predicting the starting values of the solution as long as they do not exceed the machine capacity (call it Y_{max}) and until the given time interval of integration is covered, thus yielding the prediction of both $\{Y_n\}$ and N. This is done through the procedure "PREDICT_AND_ESTIMATE_N", given in Appendix 1.

- **Rough Estimation of N**
  This is a simpler way that estimates N independently of the predictions.
  Based on the values $\{\beta_n\}_{n \leq n_s}$, one devises a rough estimation for β_n:

$$\forall n > n_s, \ \ \beta_n \approx \tilde{\beta}.$$

  Together with the estimation $\tilde{s}$ for s_n, this yields:

$$\tilde{T} = T_{n_s} + (N - n_s)\tilde{\beta}\tilde{s}.$$

  Thus, the following rough estimation of N:

$$N = \left\lceil n_s + \frac{T - T_{n_s}}{\tilde{\beta}\tilde{s}} \right\rceil. \tag{4.20}$$

- **Case of quasi-periodic problems:**
  In the case of quasi-periodic problems derived from the perturbation of a periodic problem, the sizes of successive time-slices differ very slightly from the common size ΔT_{per} of all slices of the corresponding periodic problem. Therefore, one could calculate ΔT_{per} (by solving one unique slice) and consider that it approximates the size of any slice:

$$\forall n, \quad \Delta T_n = T_n - T_{n-1} \approx \Delta T_{per}.$$

  The following estimation of the total number of slices can then be deduced:

$$N = \left\lceil \frac{T}{\Delta T_{per}} \right\rceil. \tag{4.21}$$

Remark 7 *1. In the numerical experiments of this thesis, no refined estimation of N is done. In Chapters 5 and 6, the problems are explosive and we have chosen to solve them as long as the solution is not exceeding the machine capacity. Whereas in Chapter 7, the problem is quasi-periodic and N has been approximately determined by (4.21), for a given interval $[0, T]$.*

2. In all cases, a rough estimation should be enough since it is updated after each iteration, when the end-of-slices times $\{T_n\}$ of the slices that have converged are available and can be compared to T.
Besides, by letting the rough estimation of N be oversized , a simple additional condition can stop the computations as soon as T is reached. The disadvantage then, is to evaluate some unnecessary predictions (however not costly).

4.4 Parallel Solving of the Rescaled systems

4.4.1 Predicted Rescaled Systems

In this section, we consider any k^{th} iteration and we omit the superscript $^{(k)}$ that usually designates the predicted values associated with this k^{th} iteration.

Each n^{th} rescaled initial value shooting problem ($n > n_s$) is solved, starting with a predicted value Y^p_{n-1} that has the effect of substituting β^p_n, D^p_n and G^p_n for β_n, D_n and G_n respectively. One seeks then for both the rescaled time s^c_n and the rescaled function $Z^c_n : [0, s_n] \to \mathbb{R}^K$, such that:

$$\begin{cases} \frac{dZ^c_n}{ds} = G^p_n(Z^c_n), & 0 < s \leq s^c_n, & (4.22.1) \\ Z^c_n(0) = 0, & & (4.22.2) \\ H\left[Z^c_n(s)\right] \neq 0, \ \forall s < s^c_n \ \text{ and } \ H\left[Z^c_n(s^c_n)\right] = 0, & & (4.22.3) \end{cases} \tag{4.22}$$

where $G^p_n(Z^c_n) = \beta^p_n (D^p_n)^{-1} F\left(Y^p_{n-1} + D^p_n Z^c_n\right)$.
The numerical integration yields the corrected rescaled end-of-slice values s^c_n and $Z^c_n\left(s^c_n\right)$ of which are deduced the corrected end-of-slice values Y^c_n and T^c_n:

$$Y^c_n = Y^p_{n-1} + D^p_n Z^c_n(s^c_n), \tag{4.23}$$

$$T^c_n = T^c_{n-1} + \beta^p_n s^c_n. \tag{4.24}$$

Note that the time T_n is never predicted: it is calculated as convergence progresses.

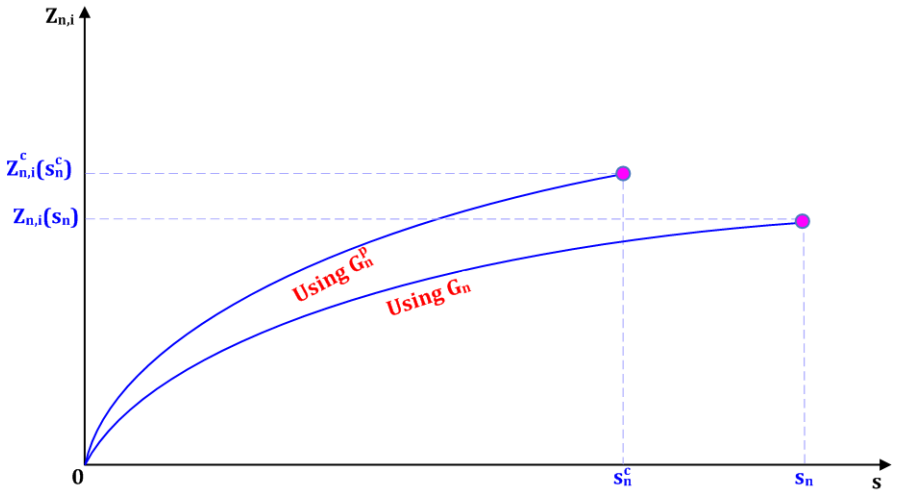


Figure 4.3: RaPTI Algorithm - Gaps Between exact and Corrected Values

4.4.2 Convergence Analysis

All our numerical experiments have shown that the corrected values $Z^c_n(.)$ of the rescaled solution, as well as the corrected end-of-slice rescaled times s^c_n, are very close to their exact values (i.e. sequentially computed) when the predicted starting values are good enough.
So far, we did not succeed in proving this important result that is essential for devising our test of convergence.
This motivates the following assumption.

Assumption 2 : *Convergence of* $Z_n^c(.)$ *and* s_n^c
The corrected solution $(s_n^c, Z_n^c(.))$ *of problem (4.22) converges to the exact solution* $(s_n, Z_n(.))$ *of problem* (S'_n)*:*

$$\begin{array}{l} \forall \epsilon_1, \epsilon_2 > 0, \ \exists \epsilon > 0, \\ \left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \epsilon \implies \left\{ \begin{array}{l} \left\|Z_n^c\left(s_n^c\right) - Z_n\left(s_n\right)\right\|_\infty \leq \epsilon_1, \\ \left|s_n^c - s_n\right| \leq \epsilon_2. \end{array} \right. \end{array} \tag{4.25}$$

Theorem 6 : *Convergence of* Y_n^c *and* T_n^c

1. *Under assumption 2, if one has* $\left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \epsilon$*, where* $\epsilon = \psi(\epsilon_1, \epsilon_2)$*, then:*

$$\left\|Y_n^c - Y_n\right\|_\infty \leq \epsilon_3, \tag{4.26}$$

with $\epsilon_3 = \epsilon_1 \left\|D_n\right\|_\infty + \epsilon \left\|Z_n\left(s_n\right)\right\|_\infty + \epsilon + \epsilon\epsilon_1$.

2. *If also* $\left|T_{n-1}^c - T_{n-1}\right| \leq \epsilon_4$ *and* β_n *is a continuous function of* Y_{n-1} *verifying a Lipschitz condition (of constant* ν_2*), then:*

$$\left|T_n^c - T_n\right| \leq \epsilon_5, \tag{4.27}$$

with $\epsilon_5 = \epsilon_4 + \beta_n\epsilon_2 + s_n\nu_2\epsilon + \nu_2\epsilon\epsilon_2$.

Proof.

- Assume $\left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \epsilon$, where $\epsilon = \psi(\epsilon_1, \epsilon_2)$, implying, by assumption 2, that: $\left\|Z_n^c\left(s_n^c\right) - Z_n\left(s_n\right)\right\|_\infty \leq \epsilon_1$.
  Then, there exist vectors ϵ_v, $\epsilon_{v_1} \in \mathbb{R}^K$ such that:
  $Y_{n-1}^p = Y_{n-1} + \epsilon_v$, with $\left\|\epsilon_{v_1}\right\|_\infty \leq \epsilon$, and
  $Z_n^c\left(s_n^c\right) = Z_n\left(s_n\right) + \epsilon_{v_1}$, with $\left\|\epsilon_{v_1}\right\|_\infty \leq \epsilon_1$.
  This makes $D_n^p = D_n + D_{\epsilon_v}$ where D_{ϵ_v} is a diagonal matrix having ϵ_v on its main diagonal, with $\left\|D_{\epsilon_v}\right\|_\infty = \left\|\epsilon_v\right\|_\infty \leq \epsilon$.
  Since $Y_n^c = Y_{n-1}^p + D_n^p Z_n^c\left(s_n^c\right)$. It follows:
  $$\begin{array}{rl} Y_n^c & = Y_{n-1} + \epsilon_v + \left[D_n + D_{\epsilon_v}\right]\left[Z_n\left(s_n\right) + \epsilon_{v_1}\right] \\ & = Y_{n-1} + D_n Z_n\left(s_n\right) + \epsilon_1\left\|D_n\right\|_\infty + \epsilon\left\|Z_n\left(s_n\right)\right\|_\infty + \epsilon + \epsilon\epsilon_1, \end{array}$$
  yielding:
  $$Y_n^c = Y_n + \epsilon_{v_c},$$
  where $\epsilon_{v_c} = \epsilon_1\left\|D_n\right\|_\infty + \epsilon\left\|Z_n\left(s_n\right)\right\|_\infty + \epsilon + \epsilon\epsilon_1 \in \mathbb{R}^K$ is such that:
  $\left\|\epsilon_{v_c}\right\|_\infty \leq \epsilon_1\left\|D_n\right\|_\infty + \epsilon\left\|Z_n\left(s_n\right)\right\|_\infty + \epsilon + \epsilon\epsilon_1 = \epsilon_3$.
  One concludes then: $\left\|Y_n^c - Y_n\right\|_\infty \leq \epsilon_3$.

- Assume $\left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \epsilon$, where $\epsilon = \psi(\epsilon_1, \epsilon_2)$, implying, by assumption 2, that: $\left|s_n^c - s_n\right| \leq \epsilon_2$.
  Then there exists $\epsilon_a \in \mathbb{R}$, such that: $s_n^c = s_n + \epsilon_a$, with $\left|\epsilon_a\right| \leq \epsilon_2$.
  On the other hand, since β_n is a continuous function of Y_{n-1} verifying a Lipshitz condition (of constant ν_2), one has:
  $\left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \epsilon \Longrightarrow \left|\beta_n^p - \beta_n\right| \leq \nu_2\left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \nu_2\epsilon$,
  making $\beta_n^p = \beta_n + \epsilon_b$, where $\epsilon_b \in \mathbb{R}$ satisfies: $\left|\epsilon_b\right| \leq \nu_2\epsilon$.
  But $\left|T_n^c - T_n\right| = \left|T_{n-1}^c + \beta_n^p s_n^c - T_{n-1} - \beta_n s_n\right|$. It follows:
  $\left|T_n^c - T_n\right| \leq \left|T_{n-1}^c - T_{n-1}\right| + \left|\beta_n^p s_n^c - \beta_n s_n\right| \Longleftrightarrow$
  $\left|T_n^c - T_n\right| \leq \left|T_{n-1}^c - T_{n-1}\right| + \left|\left(\beta_n + \epsilon_b\right)\left(s_n + \epsilon_a\right) - \beta_n s_n\right| \Longleftrightarrow$
  $\left|T_n^c - T_n\right| \leq \left|T_{n-1}^c - T_{n-1}\right| + \left|\beta_n\epsilon_a + \epsilon_b s_n + \epsilon_a\epsilon_b\right| \Longleftrightarrow$
  $\left|T_n^c - T_n\right| \leq \epsilon_4 + \beta_n\epsilon_2 + s_n\nu_2\epsilon + \nu_2\epsilon\epsilon_2 = \epsilon_5$.

■

Note that, in case of explosive solution, ϵ_1, ϵ_2 and therefore ϵ, should be sufficiently small for ϵ_3 and ϵ_5 to be small enough.

Corollary 6 ***: Test of Convergence***
Under assumption 2, if one has $\left\|Y_{n-1}^p - Y_{n-1}\right\|_\infty \leq \epsilon$, *where* $\epsilon = \psi(\epsilon_1, \epsilon_2)$, *and if:*

$$\|Y_n^p - Y_n^c\|_\infty \leq \epsilon_6, \tag{4.28}$$

then:

$$\|Y_n^p - Y_n\|_\infty \leq \epsilon_7,$$

with $\epsilon_7 = \epsilon_3 + \epsilon_6$.

Proof.
$\|Y_n^p - Y_n\|_\infty = \|Y_n^p - Y_n^c + Y_n^c - Y_n\|_\infty \leq \|Y_n^p - Y_n^c\|_\infty + \|Y_n^c - Y_n\|_\infty$.
Using theorem 6, one deduces: $\|Y_n^p - Y_n\|_\infty \leq \epsilon_6 + \epsilon_4 = \epsilon_7$ ■

The previous results can be understood as follows:

- if the predicted value Y_{n-1}^p is accurate enough, then the corrected values Y_n^c and T_n^c are close enough to the exact values and can approximate them.
  This n^{th} slice is considered to have converged.
- if also Y_n^p is close enough to Y_n^c, then the predictions are considered to keep being accurate enough and could be used for the correction of the next slice number $n+1$.

Since one starts with exact predicted values Y_{n_s} and T_{n_s}, at slice number $n_s + 1$, those results can be applied recurrently and allow to say that the slices will converge as long as the test of convergence $Y_n^p \approx Y_n^c$ holds.

4.5 RaPTI Algorithm

4.5.1 Overview of Parallel Implementation

1. **Cyclic Distribution:**
   Recall that the first n_s slices are solved sequentially (by all processors).
   Once this initialization stage of the algorithm ends, the remaining time-slices ($n > n_s$) are statically allocated to the different processors, based on a cyclic distribution. If there are n_p active processors, the j^{th} processor will be assigned slices number n, where:
   $$n = n_s + j + ln_p, \tag{4.29}$$
   with $l \in \{0, 1, \cdots \left\lfloor \frac{N-n_s-j}{n_p} \right\rfloor\}$, meaning that $(n - n_s)$ is congruent to j mod n_p.

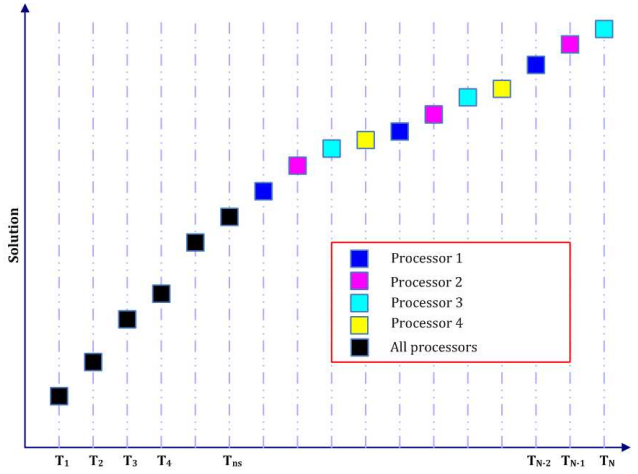


Figure 4.4: RaPTI Algorithm - Cyclic Distribution of Time Slices

Such choice of a cyclic distribution, besides its simplicity, has the advantage of facilitating the synchronization of the parallel computations, as it will be detailed in the correction procedure, thus providing a *load balanced distribution* of the work.

2. **Duplication Approach:**
   Duplication consists in having sequential procedures, within a parallel algorithm, be executed simultaneously on all processors. This avoids the communications resulting from letting only one processor execute the sequential part and then transmit its results to the other processors. In this way, communication overhead would decrease, reducing the time of the parallel execution, and hence increasing speed-up and efficiency. Furthermore, idle time for any processor would be avoided.

   Note that in the case of RaPTI Algorithm, the sequential part consists of the prediction steps (the first prediction step including the initialization stage). Figure (4.5) illustrates the difference between the two approaches.

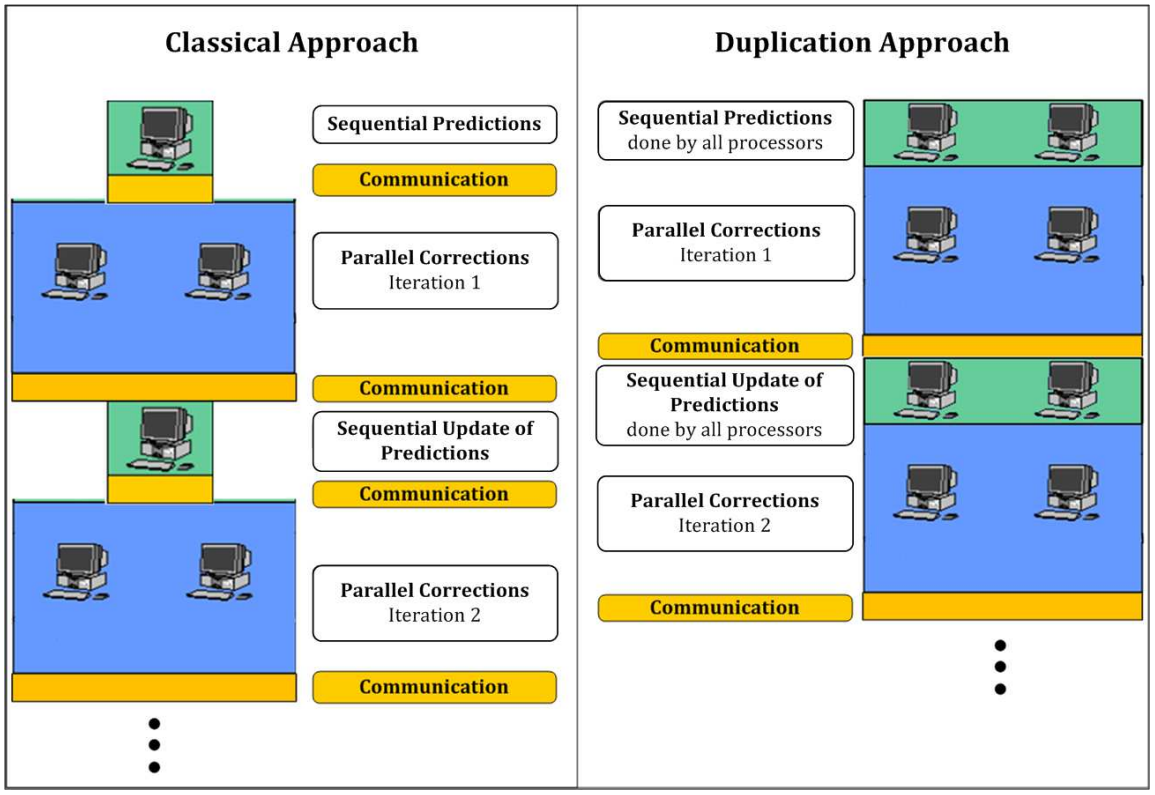


Figure 4.5: RaPTI Algorithm - Classical approach v/s Duplication Approach

In a classical implementation as that displayed in the left part of Figure (4.5), one processor predicts the starting values and sends them to all other processors, which in turn solve in parallel for the solution at each time-slice assigned to each of them. Subsequently, they send the values needed for the master processor to do the predictions and wait until receiving from the latter another set of predicted values. This process is iterated until all slices are solved.
The duplication approach is exhibited in the right part of Figure (4.5): the same sequential work of prediction is being executed on all processors, thus reducing the communications to one communication step at the end of each iteration.

3. **Overview of RaPTI Algorithm:**

Once the preliminary analysis is done, the parallel implementation of RaPTI algorithm is divided into 2 major steps, as detailed below.

RaPTI Algorithm

- **Step 1:** **Initialize**
  - *Reach the Ratio Property* (rf. section 4.5.2):
    Sequential computations, duplicated on all processors and yielding a number n_s of slices at which the ratio property is reached, up to $\epsilon_{tol}^{n_s}$, and the problem completely solved by all processors.
  - *Initialize* the number of iteration at $k = 1$, and the number of slices having converged before the first iteration at $n_s^{(k-1)} = n_s$.
- **Step 2:** **Iterate**
  - **Step 2.1: Predict** (rf. section 4.5.3)
    - *Update the parameters of the mathematical model*:
      Sequential computations, duplicated on all processors.
    - *Predict*: Sequential computations, duplicated on all processors, making every processor compute the predictions on all remaining slices ($n_s^k < n \leq N$), using the updated model.
  - **Step 2.2: Correct** (rf. section 4.5.4)
    Parallel computations: each processor solves its slices as long as not any slice diverges.
  - **Step 2.3: Communicate** (rf. section 4.5.5)
    Communication step, intended to find the number $n_s^{(k)}$ of the last converging slice at the k^{th} iteration, the times $\{T_n^c\}_{n_s^{(k-1)} < n \leq n_s^{(k)}}$ and allowing, if $T^c_{n_s^{(k)}} < T$, to communicate the last n_{prec} values of Y_n.
  - **Step 2.4:**
    If $T^c_{n_s^{(k)}} < T$, let $k = k + 1$ and repeat Step (2) until $T^c_{n_s^{(k)}} \geq T$.

Conclude
Each processor displays the solution $Y(t)$ of the slices it has computed.

Figure (4.6) displays the various steps of RaPTI algorithm.

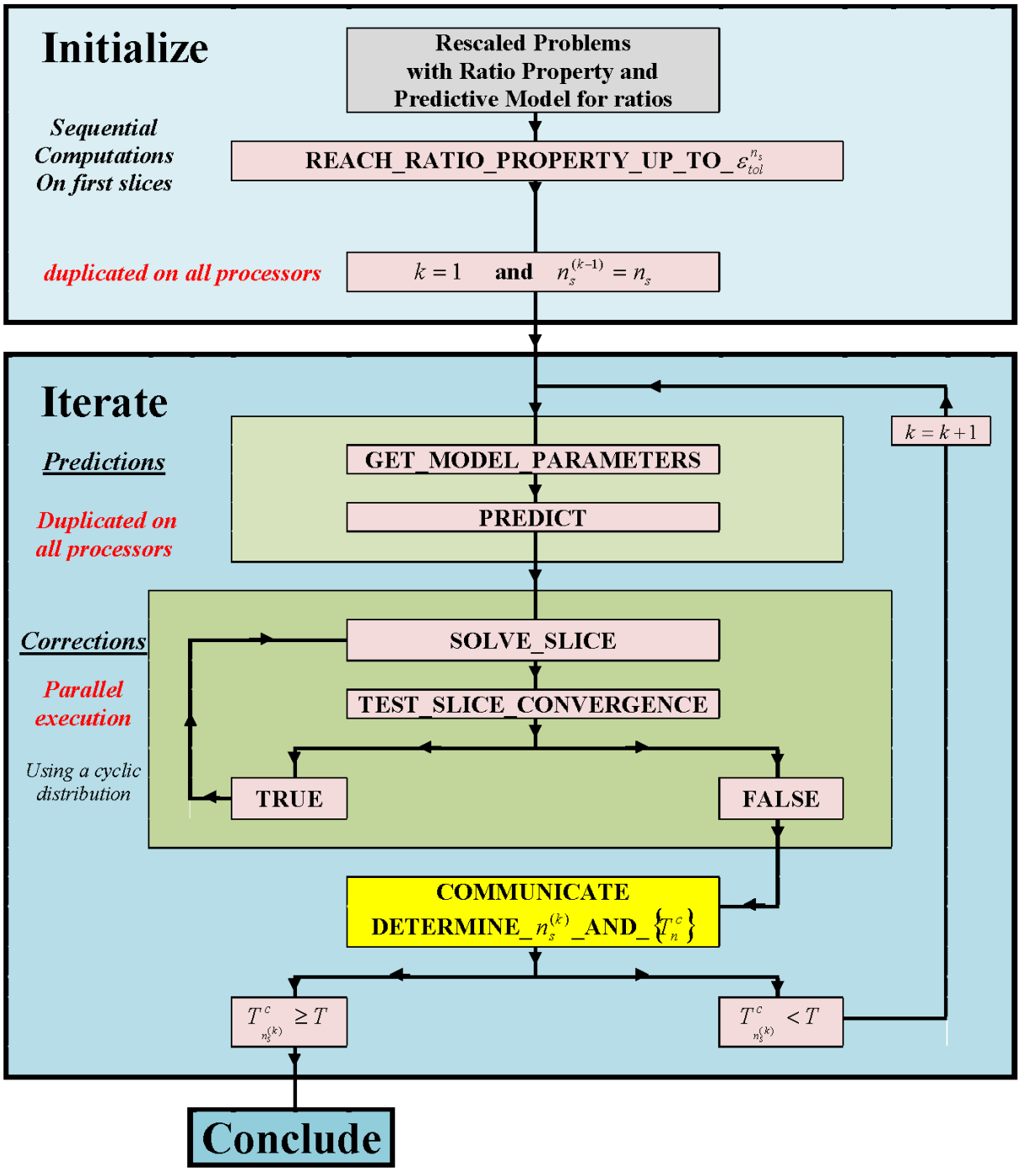


Figure 4.6: RaPTI Algorithm - Organigram.

All the indicated procedures are detailed in the next subsections.

4.5.2 Procedures for Reaching the Ratio Property

The initialization part of RaPTI algorithm consists mainly in a sequential computation intended to reach numerically the ratio property, up to a tolerance $\epsilon_{tol}^{n_s}$, providing n_s and a set of exact (sequentially computed) ratios that is necessary for updating the model.
This is redundantly implemented on all processors and uses a procedure REACH_RATIO_PROPERTY_AND_DETERMINE_n_s that is exactly similar to that used in the preliminary analysis, after letting $\epsilon = \epsilon_{tol}^{n_s}$:

- In case of weak similarity, it is similar to the procedure: DETECT_RATIO_PROPERTY_UP_TO_ϵ.
- In case of asymptotic similarity, it could be similar to one of the procedures: DETECT_RATIO_PROPERTY_UP_TO_ϵ, or REACH_ASYMPTOTIC_RATIO_PROPERTY_UP_TO_ϵ.

4.5.3 Procedures for Predictions

Ultimately, the procedure FIND_RATIOS_MODEL, described in section 4.3.2 and providing a mathematical model for predicting the ratios, could be repeated at each iteration of the predictor-corrector scheme.
However, the numerical experiments show that the common EOS condition and choice of $\{\beta_n\}$ used in the rescaling technique, and the resulting similarity and ratio properties, make the same mathematical form of the model R_{fit}, with the same number n_{prec} of previous data to be used, hold the extrapolation test at all iterations.
Hence, the procedure FIND_RATIOS_MODEL is implemented for a given problem, once for all, in the preliminary stage.

Note first that the prediction steps are duplicated on all processors.
For getting the new predictions, at the beginning of every k^{th} iteration, one should do the following:

1. *Update the parameters of the mathematical model R_{fit}*:
   The procedure "GET_MODEL_PARAMETERS", given in appendix 1, reevaluates the model's parameters that fit best the last n_{prec} ratio-vectors, in the least-squares sense, yielding the updated model $R_{fit}^{(k)}$.
2. *Predict*:
   The procedure "PREDICT", given in appendix 1, uses the mathematical model $R_{fit}^{(k)}$ for providing (by extrapolation) the predicted next ratios $\{R_n^p\}_{n>n_s^{(k)}}$, necessary to the k^{th} iteration:
   $$\forall n > n_s^{(k)}, \qquad R_n^{p^{(k)}} = R_{fit}^{(k)}(n). \tag{4.30}$$
   The predicted next starting values $\{Y_n\}_{n>n_s^{(k)}}$ of the solution are therefore obtained, using the last calculated value $Y^c_{n_s^{(k)}}$ and the recurrence (4.6):
   $$\forall n > n_s^{(k)}, \qquad Y_n^{p^{(k)}} = D_{R_n^{p^{(k)}}} \cdots D_{R_{n_s+1}^{p^{(k)}}} Y^c_{n_s^{(k)}}. \tag{4.31}$$
   At this point, one should also update the estimation of the total number of slices to be solved for covering the given interval of integration $[0, T]$.
   In case one desires a refined estimation of N, then the above procedure is substituted by the procedure PREDICT_AND_ESTIMATE_N, described in section 4.3.3.

4.5.4 Procedures for Corrections

The corrections are done in parallel after each prediction step.
Recall that the slices assigned to each processor have been defined, once for all after the initialization stage, by means of a cyclic distribution.

At each k^{th} iteration, and for $n > n_s^{(k-1)}$, each j^{th} processor starts solving its slices. Let $n_{first}^{(k)}(j)$ be the first slice assigned to it, at this point. The processor solves then slices number:

$$n \in \{n_{first}^{(k)}(j)\,,\, n_{first}^{(k)}(j) + n_p\,,\, n_{first}^{(k)}(j) + 2n_p\,,\, \cdots\,,\, n_{first}^{(k)}(j) + l_{div}^{(k)} n_p\},$$

using starting predicted values $Y_{n-1}^{p^{(k)}}$, and evaluates the relative gap between the corrected end-of-slice value Y_n^c and the value $Y_n^{p^{(k)}}$ that it has previously predicted:

$$GAP_n = \frac{Y_n^c - Y_n^{p^{(k)}}}{\max\left(\|Y_n^c\|_\infty\,,\left\|Y_n^{p^{(k)}}\right\|_\infty\right)}. \tag{4.32}$$

It keeps solving as long as $n \leq N$ and the test of convergence holds:

$$\|GAP_n\|_\infty \leq \epsilon_{tol}^g, \tag{4.33}$$

where ϵ_{tol}^g is a given tolerance.
The first divergence $\left(\|GAP_n\|_\infty > \epsilon_{tol}^g\right)$ occurs at the last solved slice, i.e. at slice number $n_{div}^{(k)}(j) = n_{first}^{(k)}(j) + l_{div}^{(k)} n_p$.

When all the processors stop solving, one can get the first slice to have globally diverged, at the k^{th} iteration:

$$n_s^{(k)} = \left(\min_{1 \leq j \leq n_p} n_{div}^{(k)}(j)\right). \tag{4.34}$$

According to theorem 6 and its corollary, all the slices number:

$$n \in \{n_s^{(k-1)} + 1, n_s^{(k-1)} + 2, \cdots, n_s^{(k)}\},$$

are guaranteed to have correct starting values $\{Y_{n-1}^{p^{(k)}}\}$ and to yield correct end-of-slice values $\{Y_n^c\}$.
Besides, since the iteration is starting with correct time $T^c_{n_s^{(k-1)}}$, the calculated times $\{T_n^c\}$ are also correct (according to the same theorem).
Hence, at the end of the k^{th} iteration, one has the time-slices completely solved, until slice number $n_s^{(k)}$.

Note that there is at least one converging time-slice at each iteration, namely slice number $n_s^{(k-1)} + 1$, which is starting with an exact (not predicted) value of the solution. It follows, like for all similar schemes:

$$n_s^{(k)} \geq n_s^{(k-1)} + 1.$$

Theoretically, if the test of convergence holds and before it proceeds to solve the next of its slices, each processor should communicate after solving each n^{th} slice, for confirming the accuracy of Y_{n-1}^p (that is insured by the processor that is solving the previous slice), in

order to validate Y_n^c and Y_n^p (according to theorem 6 and its corollary), before it proceeds to solve the next of its slices, i.e. slice number $n + n_p$.
Practically, the cyclic distribution of the time slices makes the predictions stop being accurate at about the same point for all processors, i.e. on consecutive slices, creating a kind of synchronization of the parallel computations, and making such communication useless. The only risk is that a processor solves an additional slice at each iteration, however saving a lot of communications!

One should note, at this point, that RaPTI algorithm has the characteristic ability of letting each processor stop the computations as soon as one of its slices does not converge.

The correction step is done by means of the procedure:
SOLVE_MY_SLICES_IN_PARALLEL, given in appendix 1.

4.5.5 Communication Step

Globally, the duplication approach that is used reduces the number of communications to one communication step at the end of every iteration. Each of these steps consists in what follows.

1. At the end of the correction step of each k^{th} iteration, every processor j ($2 \leq j \leq n_p$) sends to processor 1 (master processor) the number $n_{div}^{(k)}(j)$ that he got and the sizes $\{\Delta T_n^c = \beta_n^p s_n^c\}$ of its time-slices that have been solved in the current iteration.

2. The master processor receives the data and evaluates the following:

   - The number $n_s^{(k)}$ of the last slice having globally converged (on all processors), given in (4.34):
   $$n_s^{(k)} = \left(\min_{1 \leq j \leq n_p} n_{div}^{(k)}(p) \right).$$
   - The starting times $\{T_n^c\}$, for $n_s^{(k-1)} + 1 \leq n \leq n_s^{(k)}$, using the recurrence:
   $$T_n^c = T_{n-1}^c + \Delta T_n^c,$$
   and starting with $T^c_{n_s^{(k-1)}}$ that is available from the previous iteration.

   This is done by means of the procedure called DETERMINE_$n_s^{(k)}$_AND_$\{T_n^c\}$, given in appendix 1.
   Once these computations done, the master processor sends the results to all other processors.

3. All processors j ($j \neq 1$) receive $n_s^{(k)}$ and $\{T_n^c\}_{n \in \{n_s^{(k-1)}+1, \cdots, n_s^{(k)}\}}$ and do the following:

   - Deduce the scope $n_{conv}^{(k)}$ of the k^{th} iteration, i.e. the number of slices globally solved during this iteration:
   $$n_{conv}^{(k)} = n_s^{(k)} - n_s^{(k-1)}. \tag{4.35}$$

- Evaluate the real time-vector of its solution on each of its computed slices in the range $n \in \{n_s^{(k-1)}, \cdots, n_s^{(k)}\}$:

$$\forall t \in [T_{n-1}^c, T_n^c], \quad t = T_{n-1}^c + \beta_n^p s_n^c.$$

- – If $T_{n_s^{(k)}} < T$, all processors update the estimation of N (getting $N^{(k)}$), based on $T_{n_s^{(k)}}$. Besides, each processor having one of the last n_{prec} slices sends to all other processors the end-of-slice value Y_n^c of the solution on those slices, since they are needed for the new predictions.
  – If $T_{n_s^{(k)}} \geq T$, then $N^{(k)} = n_s^{(k)}$ and the iterative process stops.

Figure (4.7) displays the communication step that is done after each iteration.

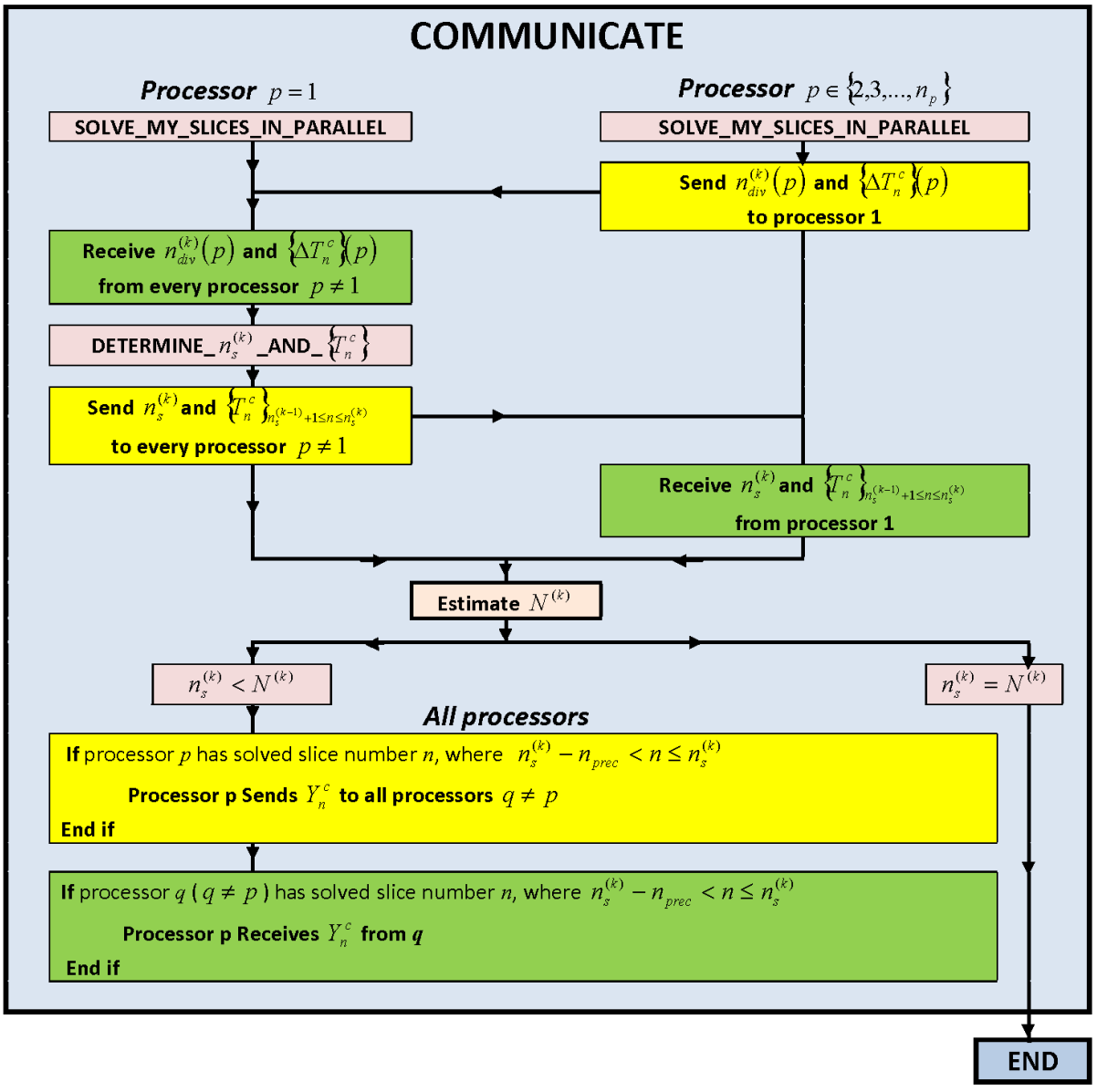


Figure 4.7: RaPTI Algorithm - Communication step

4.6 About the Numerical Integration

Solving each of the rescaled systems (S'_n) necessitates first the preliminary determination of a time-step to be used within each n^{th} slice. Then, a numerical method should be chosen for solving the initial value problem and progressing with the rescaled solution until reaching accurately the EOS condition.

4.6.1 Choice of an Initial Time-Step τ_0

For solving the rescaled initial value shooting problem (S'_n), on the n^{th} slice, using a numerical method of order r, an initial rescaled time-step is chosen according to a fixed computational tolerance ε^{τ}_{tol} up to which the accuracy is desired. This is done by means of the following iterative process:,

1. Initialize $\tau = \frac{1}{2}$, an arbitrary time-step.
2. Advance (from $s = 0$) with the solution in two ways:
   - *using one step of size* τ: the chosen numerical method yields, at $s = \tau$, an approximation W_τ of the solution $Z(\tau)$, satisfying:
   $$Z(\tau) = W_\tau + a_1\tau^r + O(\tau^{r+1}) \ , \tag{4.36}$$
   - *using two steps of size* $\tau/2$: the same numerical method yields, at $s = \tau$, an approximation $W_{\frac{\tau}{2}}$ of the solution $Z(\tau)$, satisfying:
   $$Z(\tau) = W_{\frac{\tau}{2}} + a_1(\frac{\tau}{2})^r + O(\tau^{r+1}). \tag{4.37}$$

   Multiplying (4.37) by 2^r and then subtracting (4.36) yields:
   $$Z(\tau) = W_{\frac{\tau}{2}} + \frac{W_{\frac{\tau}{2}} - W_\tau}{2^r - 1} + O(\tau^{r+1}) \ . \tag{4.38}$$
3. Repeat the previous step after substituting $\tau/2$ for τ, until reaching a time-step τ_0 satisfying:
   $$\left\| \frac{W_{\frac{\tau_0}{2}} - W_{\tau_0}}{W_{\frac{\tau_0}{2}}} \right\|_\infty \leq (2^r - 1)\ \varepsilon^{\tau}_{tol} \leq \left\| \frac{W_{\tau_0} - W_{2\tau_0}}{W_{\tau_0}} \right\|_\infty . \tag{4.39}$$

Note that, because of the similarity of the rescaled systems, this choice of initial time-step τ_0 can be done once for all, before solving the first slice, and be used for all slices.

4.6.2 Choice of a Numerical Method

As a consequence to the local approach for solving each n^{th} slice, resulting from the rescaling methodology, an explicit method is used in this thesis, namely: the very popular explicit Runge-Kutta Method of order 4, which is commonly referred to as RK4. It is well-known that the RK4 method is a fourth-order method, meaning that the error per step is on the order of τ^5, while the total accumulated error has order τ^4.
The numerical method leads, on the rescaled time interval $[0, s_n]$, a sequence $\{Z_{n,j} \,|\, j = 1, \cdots, m\}$ that approximates the solution $\{Z_n(j\tau) \,|\, j = 1, \cdots, m\}$, where the time-step τ of the interval of integration $[0, s_n]$ is a priori selected on the basis of a computational tolerance, as indicated above.
Since it is clear that, in this context, one is working in the n^{th} slice, the notation Z_j is substituted to $Z_{n,j}$.
The RK4 method for solving this problem starts with $Z_0 = 0$, at $s_0 = 0$, and evaluates recurrently Z_{j+1} at $s_{j+1} = s_j + \tau$, in terms of Z_j and τ as follows:
$K_1 = G_n(Z_j)$
$K_2 = G_n(Z_j + \frac{1}{2}\tau K_1)$
$K_3 = G_n(Z_j + \frac{1}{2}\tau K_2)$
$K_4 = G_n(Z_j + \tau K_3)$
$Z_{j+1} = Z_j + \frac{1}{6}\tau(K_1 + 2K_2 + 2K_3 + K_4)$, at $s_{j+1} = s_j + \tau$.

4.6.3 Reaching accurately the EOS condition

For a chosen numerical method, the problem is solved on the n^{th} slice, with a constant rescaled time-step τ, yielding the value Z_{j+1} of the solution at time s_{j+1} from its value Z_j at time s_j: $Z_{j+1} = Num\,(G_n, Z_j, \tau)$.
Then, it progresses with the solution until overstepping the EOS condition.
As soon as it happens, the rescaled time-step τ is refined in order to reach accurately (up to a tolerance ε_{tol}^{eos}) the EOS condition, as follows:

1. Substitute $\tau/2$ for τ.

2. Advance with the solution, starting from the previous time instant (at which the EOS condition was not overstepped yet) and until the EOS condition is overstepped once more.

3. Repeat steps 1 and 2 while τ is larger than a desired tolerance ε_{tol}^{eos}.

This is done through the following procedures, given in appendix 1:

- "SOLVE_SLICE": procedure solving a time slice, in the sequential initialization part.

- "SOLVE_SLICE_PARALLEL": procedure solving a time slice with an unknown initial time, in the parallel correction step of the iterative process.

- "SOLVE_SLICE_LIMIT": procedure solving the slice of the limit problem.

Chapter 5

Membrane Problem

The considered membrane problem is a second order scalar initial value problem:

$$y'' - b|y'|^{q-1}\, y' \;+\; |y|^{p-1}\, y = 0,\;\; t > 0, \quad y(0) = y_{1,0}, \quad y'(0) = y_{2,0}.$$

After lowering the order, it reduces to the general form (S) of a first order initial value problem, of dimension 2.

When the problem parameters p and q are such that $0 < p \leq q \leq \frac{2p}{p+1}$, the solution on $[0, \infty)$ exhibits an explosive behavior, in an oscillatory way.
The application of the time-slicing and rescaling techniques described in Chapter 1, yields then:

- an invariance property when $q = \frac{2p}{p+1}$,
- an asymptotic similarity when $0 < p \leq q < \frac{2p}{p+1}$.

This makes RaPTI algorithm to converge very fast, yielding excellent speed-ups (evaluated as described in Appendix 2).

The notations used in this chapter are those indicated in table 10.1.

5.1 Description of the problem

Consider the second order initial value problem in which one seeks $y : [0, T] \longrightarrow \mathbb{R}$ $(T \leq \infty)$ such that:

$$\begin{cases} y'' - b|y'|^{q-1}\, y' \;+\; |y|^{p-1}\, y = 0, \;\; t > 0, & (5.1.1) \\ y(0) = y_{1,0}, & (5.1.2) \\ y'(0) = y_{2,0}. & (5.1.3) \end{cases} \qquad (5.1)$$

This model describes the motion of a membrane element linked to a spring. The non-linear term in y is related to the rigidity of the spring and that in y' models a "speed-up" of the phenomenon when $b > 0$, and a "slow-down" when $b < 0$. In this last case, the initial-value problem is dissipative and the existence of the solution is global on $[0, \infty)$ ([55], [56]).

When $b > 0$, the speed-up of the motion causes a "blow-up" of the solution, i.e. the existence of $T_b \leq \infty$, such that: $\lim_{t \to T_b} |y(t)| = \lim_{t \to T_b} |y'(t)| = \infty$.
This case has been intensively studied by Souplet et al in [57], [58], [59] and [60], where the existence and uniqueness of the solution have been discussed and the behavior of the solution determined in some cases: in the case where $b = 1$ and $p, q > 1$, they proved the existence of two critical values $q = p$ and $q = \frac{2p}{p+1}$ determining, in the (p, q) plane, three regions of different behaviors. Figure 5.1 and table 5.1 summarize those results.

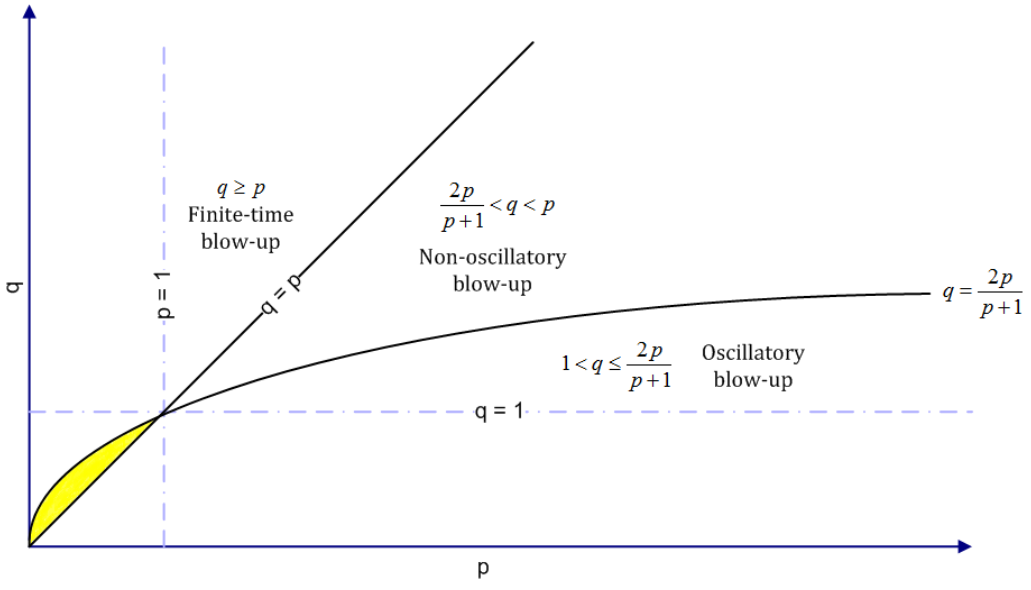


Figure 5.1: Membrane Problem - Different Behaviors, when $b = 1$ and $p, q > 1$.

p,q	**Behavior of the Solution**
$1 < p \leq q$	Finite-time blow-up
$1 < \frac{2p}{p+1} < q < p$	Non-oscillatory blow-up
$1 < q \leq \frac{2p}{p+1}$	Oscillatory blow-up

Table 5.1: Membrane Problem - Different Behaviors when $b = 1$ and $p, q > 1$.

In [49] and [51], we have successfully applied the rescaled methodology to the case where $p > 1$ and $q = \frac{2p}{p+1}$, of which the solution is exploding in finite time, in an oscillatory or non oscillatory way (depending on the value of b). It was done in the purpose of controlling the stiffness of this explosive problem and resulted in getting very accurate values of the solution and of the finite time of explosion.

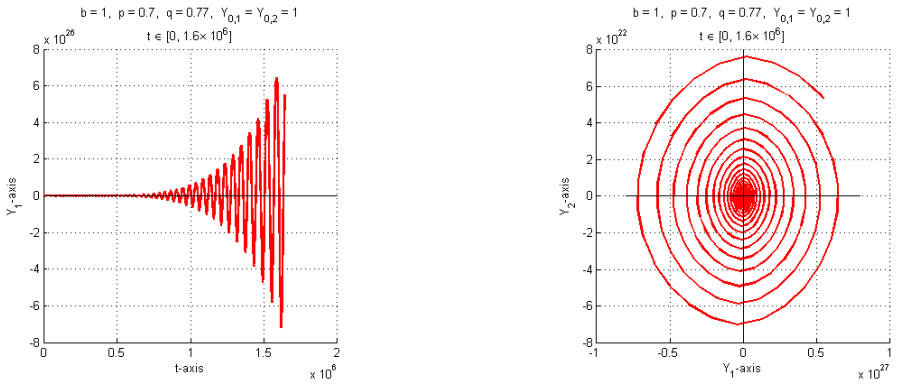
In this chapter, we consider the case:

$$0 < p \leq q \leq \frac{2p}{p+1} \leq 1, \tag{5.2}$$

that is represented by the colored area of figure 5.1.
For a pair (p, q) verifying this condition, we did not find (at this point) any reference proving the existence and uniqueness of a solution, neither its behavior.
However, the numerical integration of (5.1) have shown an explosive behavior, with an infinite blow-up time and both y and y' exhibiting oscillatory behaviors, i.e.:

1. $\lim_{t\to\infty} |y(t)| = \lim_{t\to\infty} |y'(t)| = \infty$, and
2. $y(t)$ and $y'(t)$ admit an infinite number of roots in the interval $[0, \infty)$.

We assume, in this chapter, that the behavior of the solution is indeed explosive, in an oscillatory way, for all p and q satisfying (5.2).
Such behavior makes the solution, in the phase-plane (y, y'), spiral outwards about the origin toward infinity.



(a) Y versus t (b) Phase plane

Figure 5.2: Membrane Problem - Global Behavior, when $0 < p \leq q \leq \frac{2p}{p+1} \leq 1$.

Our first step is to write (5.1) as a system of first order ODE's.
Letting $Y_1 = y$ and $Y_2 = y'$ makes problem (5.1) equivalent to the following system, in

which one seeks $Y_1 : [0,\infty] \longrightarrow \mathbb{R}$ and $Y_2 : [0,\infty] \longrightarrow \mathbb{R}$, such that:

$$\left\{ \begin{array}{ll} Y_1^{'}(t) = Y_2, & (5.3.1) \\ Y_2^{'}(t) = b|Y_2|^{q-1}\ Y_2\ -\ |Y_1|^{p-1}\ Y_1, & (5.3.2) \\ Y_1(0) = Y_{1,0}, & (5.3.3) \\ Y_2(0) = Y_{2,0}. & (5.3.4) \end{array} \right. \tag{5.3}$$

Hence, problem (5.1) reduces to the first order initial value problem in which one seeks the solution $Y : [0,\infty) \to \mathbb{R}^2$, such that:

$$(S) \qquad \left\{ \begin{array}{ll} \frac{dY}{dt} = F(Y), & t > 0 \\ Y(0) = Y_0, & \end{array} \right.$$

where $Y_0 = \begin{pmatrix} Y_{1,0} \\ Y_{2,0} \end{pmatrix}$ and:

$$F \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} = \begin{pmatrix} Y_2 \\ b|Y_2|^{q-1}\ Y_2\ -\ |Y_1|^{p-1}\ Y_1 \end{pmatrix}. \tag{5.4}$$

5.2 Rescaling and Similarity Properties

5.2.1 Choice of an EOS condition

As discussed in Chapter 2, in case of oscillatory behavior of the solution and when one seeks a time-parallelism, the slicing technique would inflict that the slice be ended when the solution completes a full (or almost full) rotation. One possible way to do it is to end an n^{th} slice when:

$$\left\{ \begin{array}{lll} \text{at } t = T_n, & Y_{n,2} = |Y_{n,1}|^{\frac{p+1}{2}} & \text{with } Y_{n,1} > 0 \\ \forall t \in (T_{n-1}, T_n)\,, & Y_2(t) \neq |Y_1(t)|^{\frac{p+1}{2}} & \text{or } Y_1(t) \leq 0 \end{array} \right. \tag{5.5}$$

This EOS condition is governed by a family of functions $E_n : \mathbb{R}^2 \to \mathbb{R}$, given by:

$$\forall W \in \mathbb{R}^2, \quad E_n\left[W\right] = W_2 - |W_1|^{\frac{p+1}{2}}\,, \tag{5.6}$$

and defined by adding, to the condition $E_n\left[Y_n\right] = 0$, the additional constraint $Y_{n,1} > 0$ that is intended to make the solution complete an almost full rotation in the phase plane:

$$\left\{ \begin{array}{lll} \text{at } t = T_n, & E_n\left[Y_n\right] = 0 & \text{with } Y_{n,1} > 0 \\ \forall t \in (T_{n-1}, T_n)\,, & E_n\left[Y\right] \neq 0 & \text{or } Y_1(t) \leq 0 \end{array} \right.$$

Let L be the logical function defined by:

$$\left\{ \begin{array}{lll} L\left[Y(t)\right] = 0 & \text{if } Y_2(t) = |Y_1(t)|^{\frac{p+1}{2}} & \text{and } Y_1(t) > 0, \\ L\left[Y(t)\right] = 1 & \text{if } Y_2(t) \neq |Y_1(t)|^{\frac{p+1}{2}} & \text{or } Y_1(t) \leq 0. \end{array} \right. \tag{5.7}$$

The EOS condition (5.5) translates then to:

$$L\left[Y(T_n)\right] = 0 \qquad \text{and} \qquad \forall t \in (T_{n-1}, T_n),\ \ L\left[Y(t)\right] \neq 0. \tag{5.8}$$

In other words, a slice is ended whenever the trajectory of the solution, in the Y_1Y_2 phase plane, intersects the curve $Y_2 = |Y_1|^{\frac{p+1}{2}}$ in the first quadrant.

The oscillating behavior of the solution makes such EOS condition guaranteed to be reached (see figure 5.3).

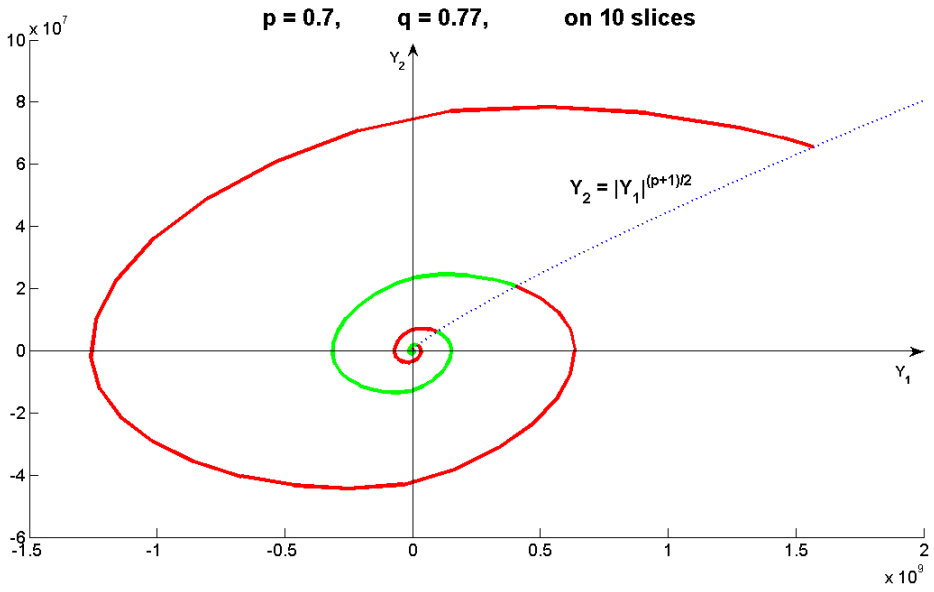
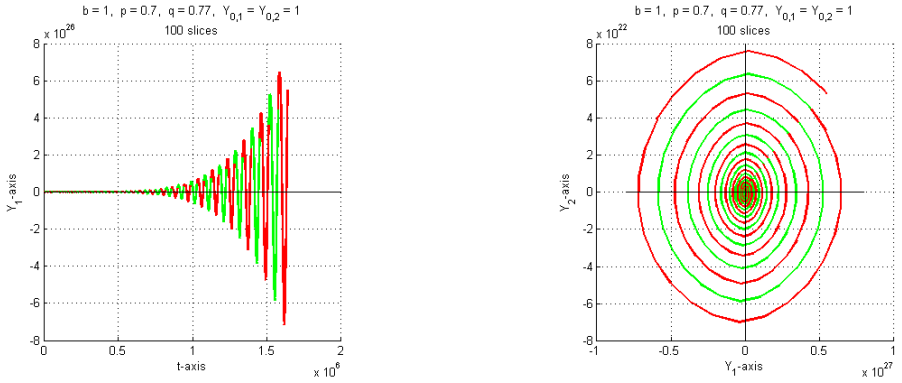


Figure 5.3: Membrane Problem - EOS Condition

This appears as a particular way for completing a rotation.
Another straightforward alternative consists in ending a slice as each time the polar angle, in the Y_1Y_2 phase plane, recovers its initial value, as described in the general choice given in Chapter 2.
However, as indicated later in this chapter, our choice leads to a proper illustration of both invariance and asymptotic similarity.



(a) Y versus t (b) Phase plane

Figure 5.4: Membrane problem - 100 slices, $b = 1$, $p = 0.7$, $q = 0.77$, $Y_0 = (1, 1)$.

Problem (5.3) is then equivalent to the sequence of initial value shooting problems in which one seeks, on each n^{th} slice ($n \geq 1$), for the time T_n and the solution $Y : [T_{n-1}, T_n] \longrightarrow \mathbb{R}^2$

such that:

$$\begin{cases} \frac{dY_1}{dt} = Y_2, & (5.9.1) \\ \frac{dY_2}{dt} = b|Y_2|^{q-1}\ Y_2\ -\ |Y_1|^{p-1}\ Y_1, & (5.9.2) \\ Y_1(T_{n-1}) = Y_{n-1,1} & (5.9.3) \\ Y_2(T_{n-1}) = Y_{n-1,2} & (5.9.4) \\ L\left[Y(T_n)\right] = 0 \quad \text{and} \quad \forall t \in (T_{n-1}, T_n),\ \ L\left[Y(t)\right] \neq 0 & (5.9.5) \end{cases} \tag{5.9}$$

5.2.2 Rescaled Systems

Note first that the EOS condition (5.5) makes the nonzeroness condition (1.11) hold at end-of-slices.
Problem (5.9) is rescaled through the change of variables (1.12) that is explicitly given in this 2-dimensional case by:

$$\begin{cases} t = T_{n-1} + \beta_n s, & (5.10.1) \\ Y_1(t) = Y_{n-1,1}(1 + Z_{n,1}(s)), & (5.10.2) \\ Y_2(t) = Y_{n-1,2}(1 + Z_{n,2}(s)), & (5.10.3) \end{cases} \tag{5.10}$$

The equality involved in the EOS condition (5.5) rescales as:
$Y_{n-1,2}(1 + Z_{n,2}(s_n)) = |Y_{n-1,1}\left(1 + Z_{n,1}(s_n)\right)|^{\frac{p+1}{2}}$, which is equivalent to:

$$1 + Z_{n,2}(s_n) = \frac{|Y_{n-1,1}|^{\frac{p+1}{2}}}{Y_{n-1,2}} |1 + Z_{n,1}(s_n)|^{\frac{p+1}{2}}.$$

Assume, without loss of generality, that the EOS condition is satisfied by the initial condition $Y_0 \forall n \geq 1,\ \frac{|Y_{n-1,1}|^{\frac{p+1}{2}}}{Y_{n-1,2}} = 1$. (If not, one can always solve a partial time-slice, until reaching the EOS condition, and initialize the time at that point).
This makes the rescaled EOS condition independent of n:

$$\begin{cases} \text{at } s = s_n, & 1 + Z_{n,2}(s_n) = |1 + Z_{n,1}(s_n)|^{\frac{p+1}{2}} & \text{with} & 1 + Z_{n,1}(s_n) > 0 \\ \forall s \in (0, s_n), & 1 + Z_{n,2}(s) \neq |1 + Z_{n,1}(s)|^{\frac{p+1}{2}} & \text{or} & 1 + Z_{n,1}(s) \leq 0 \end{cases} \tag{5.11}$$

This rescaled EOS condition is governed by the invariant function $H : \mathbb{R}^2 \to \mathbb{R}$, given by:

$$\forall W \in \mathbb{R}^2,\ \ H\left[W\right] = (1 + W_2) - |1 + W_1|^{\frac{p+1}{2}}, \tag{5.12}$$

and defined by adding, to the condition $H\left[Z_n(s_n)\right] = 0$, the additional constraint: $1 + Z_{n,1}(s_n) > 0$.
Let L_z be the logical function defined by:

$$\begin{cases} L_z\left[Z_n(s)\right] = 0 & \text{if } 1 + Z_{n,2}(s) = |1 + Z_{n,1}(s)|^{\frac{p+1}{2}} & \text{and} & 1 + Z_{n,1}(s) > 0 \\ L_z\left[Z_n(s)\right] = 1 & \text{if } 1 + Z_{n,2}(s) \neq |1 + Z_{n,1}(s)|^{\frac{p+1}{2}} & \text{or} & 1 + Z_{n,1}(s) \leq 0 \end{cases} \tag{5.13}$$

The rescaled EOS condition (5.11) translates then to:

$$L_z\left[Z_n(s_n)\right] = 0 \quad \text{and} \quad \forall s \in (0, s_n),\ \ L_z\left[Z_n(s)\right] \neq 0. \tag{5.14}$$

Besides, it follows from (5.9) and (5.10) that:
$\frac{dZ_1}{ds} = \frac{\beta_n}{Y_{n-1,1}} Y_{n-1,2}(1 + Z_2)$

$$\frac{dZ_2}{ds} = b\beta_n|Y_{n-1,2}|^{q-1}|1+Z_2|^{q-1}(1+Z_2) \\ - \ \frac{\beta_n Y_{n-1,1}}{Y_{n-1,2}}|Y_{n-1,1}|^{p-1}|1+Z_1|^{p-1}(1+Z_1).$$

By letting:

$$\begin{cases} \omega_n = \frac{\beta_n}{Y_{n-1,1}} Y_{n-1,2}, & (5.15.1) \\ \gamma_{n,1} = \frac{\beta_n Y_{n-1,1}}{Y_{n-1,2}}|Y_{n-1,1}|^{p-1}, & (5.15.2) \\ \gamma_{n,2} = \beta_n |Y_{n-1,2}|^{q-1}, & (5.15.3) \end{cases} \tag{5.15}$$

the resulting rescaled systems are:

$$\begin{cases} \frac{dZ_{n,1}}{ds} = G_{n,1}(Z_n), \qquad 0 < s \le s_n, & (5.16.1) \\ \frac{dZ_{n,2}}{ds} = G_{n,2}(Z_n), \qquad 0 < s \le s_n, & (5.16.2) \\ Z_{n,1}(0) = Z_{n,2}(0) = 0, & (5.16.3) \\ L_z\left[Z_n(s_n)\right] = 0 \quad \text{and} \quad \forall s \in (0, s_n), \ L_z\left[Z_n(s)\right] \neq 0. & (5.16.4) \end{cases} \tag{5.16}$$

where:

$$\begin{cases} G_{n,1}(Z_n) = \omega_n(1+Z_{n,2}), \\ G_{n,2}(Z_n) = b\gamma_{n,2}|1+Z_{n,2}|^{q-1}(1+Z_{n,2}) - \ \gamma_{n,1}|1+Z_{n,1}|^{p-1}(1+Z_{n,1}). \end{cases}$$

5.2.3 Critical Choice of $\{\beta_n\}$

Taking into consideration the EOS condition (5.5), in the expressions of the coefficients given in (5.15), one obtains:
$\omega_n = \gamma_{n,1} = \beta_n |Y_{n-1,1}|^{\frac{p-1}{2}}$ and $\gamma_{n,2} = \beta_n |Y_{n-1,1}|^{\frac{(p+1)(q-1)}{2}}$.
Thus, the critical choice for the time-rescaling factor

$$\beta_n = |Y_{n-1,1}|^{\frac{1-p}{2}} = |Y_{n-1,2}|^{\frac{1-p}{p+1}}, \tag{5.17}$$

makes:

$$\begin{cases} \omega_n = 1, & (5.18.1) \\ \gamma_{n,1} = 1\ , & (5.18.2) \\ \gamma_{n,2} = |Y_{n-1,1}|^{\frac{(p+1)(q-1)}{2} + \frac{1-p}{2}} = \gamma_n, & (5.18.3) \end{cases} \tag{5.18}$$

Note that the explosive behavior of $Y_{n,1}$ makes $\gamma_{n,2} = \gamma_n \le 1$.
The sequence of rescaled initial value shooting problems (5.16) is then such that, for all n:

$$\begin{cases} G_{n,1}(Z_n) = 1 + Z_{n,2}, \\ G_{n,2}(Z_n) = b\gamma_n|1+Z_{n,2}|^{q-1}(1+Z_{n,2}) - \ |1+Z_{n,1}|^{p-1}(1+Z_{n,1}), \end{cases} \tag{5.19}$$

with:

$$\gamma_n = |Y_{n-1,1}|^{\frac{p+1}{2}\left(q - \frac{2p}{p+1}\right)} \le 1.$$

5.2.4 Invariance and Asymptotic Similarity

Theorem 7

1. *If $q = \frac{2p}{p+1}$, $\forall p \le 1$, then the rescaled systems (5.16) are invariant and equivalent, for all n, to the unique initial value shooting problem in which one seeks for the rescaled*

time s_1 and for the rescaled functions $Z_1 : [0, s_1] \to \mathbb{R}^2$, such that:

$$\left\{\begin{array}{ll} \frac{dZ_{1,1}}{ds} = (1 + Z_{1,2}), \qquad 0 < s \leq s_1, & (5.20.1) \\ \frac{dZ_{1,2}}{ds} = b|1 + Z_{1,2}|^{q-1}(1 + Z_{1,2}) & \\ \qquad\qquad - |1 + Z_{1,1}|^{p-1}(1 + Z_{1,1}), & (5.20.2) \\ Z_{1,1}(0) = Z_{1,2}(0) = 0, & (5.20.3) \\ L_z\left[Z_n(s_n)\right] = 0 \quad and \quad \forall s \in (0, s_n), \;\; L_z\left[Z_n(s)\right] \neq 0. & (5.20.4) \end{array}\right. \tag{5.20}$$

2. *If $0 < p \leq q < \frac{2p}{p+1} \leq 1$, then the rescaled systems (5.16) are asymptotically similar to a limit initial value shooting problem in which one seeks for the rescaled time s_L and for the rescaled functions $Z_{L,1} : [0, s_L] \to \mathbb{R}^2$ and $Z_{L,2} : [0, s_L] \to \mathbb{R}^2$, such that:*

$$\left\{\begin{array}{ll} \frac{dZ_{L,1}}{ds} = 1 + Z_{L,2}, \qquad 0 < s \leq s_L, & (5.21.1) \\ \frac{dZ_{L,2}}{ds} = -|1 + Z_{L,1}|^{p-1}(1 + Z_{L,1}) \;, & (5.21.2) \\ Z_{L,1}(0) = Z_{L,2}(0) = 0, & (5.21.3) \\ L_z\left[Z_L(s_L)\right] = 0 \quad and \quad \forall s \in (0, s_L), \;\; L_z\left[Z_L(s)\right] \neq 0. & (5.21.4) \end{array}\right. \tag{5.21}$$

Proof:

1. If $q = \frac{2p}{p+1}$, then $\gamma_n = 1$. It follows that all the coefficients involved on the functions G_n are independent of n, and therefore:

$$\forall n \geq 1, \qquad \left\{\begin{array}{l} G_{n,1}(Z_n) = 1 + Z_{n,2}, \\ G_{n,2}(Z_n) = b|1 + Z_{n,2}|^{q-1}(1 + Z_{n,2}) - \;|1 + Z_{n,1}|^{p-1}(1 + Z_{n,1}), \end{array}\right.$$

   or equivalently:

$$\forall n, \quad G_n(.) = G_1(.).$$

   Together with the invariance of the function H governing the EOS condition, this implies the invariance of the rescaled systems, yielding the same rescaled solution $Z_n = Z_1$ and the same rescaled time-slice size $s_n = s_1$, for all n.

2. One has:
   $G_n(Z_n) = \begin{pmatrix} G_{n,1}(Z_n) \\ G_{n,2}(Z_n) \end{pmatrix} = \begin{pmatrix} 1 + Z_{n,2} \\ b\gamma_n|1 + Z_{n,2}|^{q-1}(1 + Z_{n,2}) - |1 + Z_{n,1}|^{p-1}(1 + Z_{n,1}) \end{pmatrix}$,
   and $G_L(Z_L(s)) = \begin{pmatrix} G_{L,1}(Z_L) \\ G_{L,2}(Z_L) \end{pmatrix} = \begin{pmatrix} 1 + Z_{L,2} \\ -|1 + Z_{L,1})|^{p-1}(1 + Z_{L,1}) \end{pmatrix}$.
   It follows, for all $W = \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} \in \mathbb{R}^2$:

$$G_n(W) - G_L(W) = \begin{pmatrix} 0 \\ b\gamma_n|1 + W_2|^{q-1}(1 + W_2) \end{pmatrix}.$$

   Since $\frac{p+1}{2}\left(q - \frac{2p}{p+1}\right) < 0$, the assumption of the explosive behavior of the solution implies:

$$\lim_{n\to\infty}(\gamma_n) = \lim_{n\to\infty}\left(|Y_{n-1,1}|^{\frac{p+1}{2}\left(q-\frac{2p}{p+1}\right)}\right) = 0.$$

   It follows, for all $W \in \mathbb{R}^2$:

$$\forall \rho > 0, \quad \lim_{n\to\infty}\left[\max_{W\in B_\rho}\|G_n(W) - G_L(W)\|_\infty\right] = 0,$$

Together with the invariance of the function H governing the EOS condition, this proves the asymptotic similarity of the rescaled systems (5.16) to the limit system (5.21).

■

Figure 5.5 illustrates a case of invariance: one notices the exact superposition of the rescaled solution, in the Z_1Z_2- phase plane, on all the time-slices.

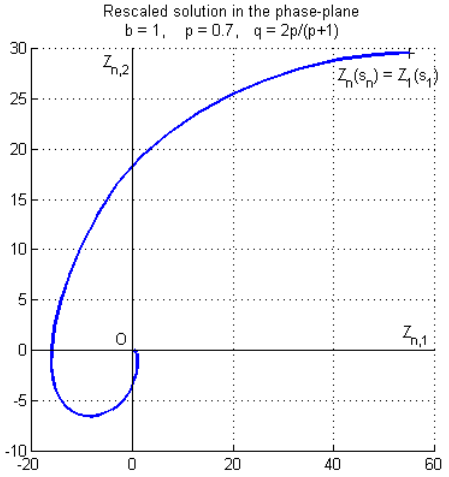


Figure 5.5: Membrane Problem - A Case of Invariance, $b = 1$, $p = 0.7$, $q = \frac{2p}{p+1}$, $Y_0 = (1, 1)$.

Figure 5.6 illustrates a case of asymptotic similarity: the rescaled solution on successive time-slices are represented on the same graph, in the Z_1Z_2- phase plane. Each trajectory, corresponding to the n^{th} slice $[0, s_n]$, starts at $Z_n(0) = (0, 0)$ and ends at $Z_n(s_n)$. One can notice how the successive trajectories becomes closer and closer to that corresponding to the limit problem (in green), as n becomes larger.

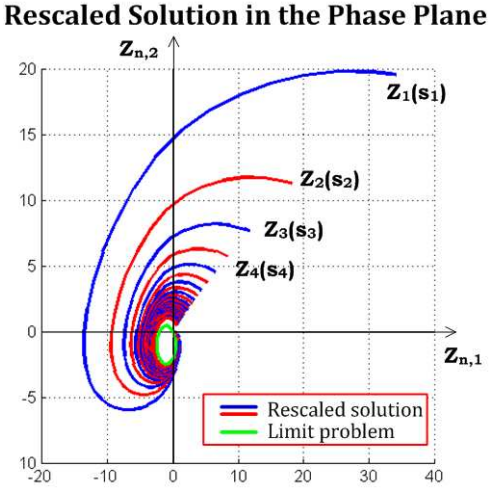


Figure 5.6: Membrane Problem - A Case of Asymptotic similarity, $b = 1$, $p = 0.7$, $q = 0.77$, $Y_0 = (1, 1)$.

5.3 Numerical Results: Case of Invariance

When $q = \frac{2p}{p+1}$, one obtains a perfect ratio property, as stated in proposition 8, i.e. the ratios of the solution on any two successive slices are constant. This constant ratio is evaluated by solving sequentially on the first slice only: $\forall n,\ R_n = R_1$.

This allows to get:
(i) the **exact** starting values $\{Y_n\}$ of the solution on all slices, using the recurrence (4.7): $\forall n, \quad Y_n = (D_{R_1})^n\, Y_0$.
(ii) the sizes of all time-slices: $\forall n,\ \Delta T_n = \beta_n s_1$, where s_1 is the invariant size of rescaled time-slices and β_n is deduced from the exact predictions of Y_n.

If parallel integration is to be done, one gets a perfect parallelism with no need of any iteration or communication.

However, and as detailed in Chapter 4, the parallelism is of no need here, since solving the problem on the first slice solves it on all slices, through a simple change of variable, using RaTI algorithm.

Table 5.2 shows some cases of invariance that are solved, using RaPTI algorithm. Even if one knows that no parallelism is needed, this is done as a bench mark, by solving all the slices, with exact predictions.
Note first that in all the considered cases, the solution is explosive and the computations stop as soon as the value of the solution exceeds the machine capacity, thus determining different number of slices, in each case, which is the maximum number.

Case	1	2	3	4	5	6	7	8	9
p	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
T_{sol}	$4.9 10^{10}$	$2.4 10^{19}$	$1.8 10^{29}$	$2.8 10^{40}$	$1.5 10^{53}$	$5.9 10^{67}$	$4 10^{84}$	$1.4 10^{104}$	$3.5 10^{127}$
N	**1160**	**1179**	**1220**	**1306**	**1323**	**1491**	**1531**	**1304**	**1932**
n_s	1	1	1	1	1	1	1	1	1
n_I	1	1	1	1	1	1	1	1	1
$\mathbf{T_s}$	**4.62**	**4.70**	**5.26**	**5.42**	**5.55**	**6.25**	**6.68**	**5.62**	**8.25**
T_2	2.22	2.23	2.50	2.56	2.60	2.97	3.18	2.71	3.85
$\mathbf{E_2}$	**1.042**	**1.051**	**1.052**	**1.059**	**1.070**	**1.051**	**1.050**	**1.036**	**1.073**
$\mathbf{S_2}$	**2.084**	**2.102**	**2.104**	**2.119**	**2.140**	**2.103**	**2.100**	**2.073**	**2.145**
S_2^{max}	1.998	1.998	1.998	1.998	1.998	1.999	1.999	1.998	1.999
T_4	1.15	1.19	1.29	1.33	1.34	1.53	1.62	1.40	1.98
$\mathbf{E_4}$	**1.003**	0.990	**1.021**	**1.020**	**1.034**	**1.020**	**1.031**	**1.003**	**1.044**
$\mathbf{S_4}$	**4.012**	**3.961**	**4.082**	**4.082**	**4.135**	**4.081**	**4.125**	**4.012**	**4.175**
S_4^{max}	3.990	3.990	3.990	3.991	3.991	3.992	3.992	3.991	3.994
T_8	0.64	0.64	0.70	0.71	0.73	0.82	0.88	0.75	1.04
$\mathbf{E_8}$	**0.908**	**0.919**	**0.935**	**0.951**	**0.954**	**0.951**	**0.954**	**0.935**	**0.996**
$\mathbf{S_8}$	**7.261**	**7.348**	**7.482**	**7.604**	**7.633**	**7.606**	**7.635**	**7.483**	**7.968**
S_8^{max}	7.952	7.953	7.954	7.957	7.958	7.963	7.964	7.957	7.971

Table 5.2: Membrane - Cases of Invariance: Application of RaPTI algorithm *by solving all slices, with exact predictions*, with $b = 1$ and $q = \frac{2p}{p+1}$.

RaPTI has been tested on the considered 9 cases, using 2, 3, 4, 5, 6, 7, and 8 processors. Figure 5.7 shows how the values of the speed-up, averaged on the 9 cases, vary with the number of processors. It shows also how close it is to the maximum speed-up, stated by Amdhal's law.

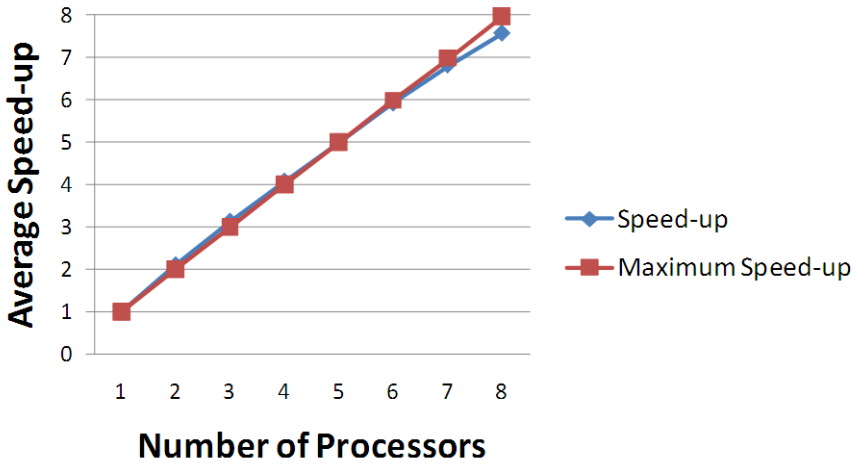


Figure 5.7: Membrane - Cases of Invariance: Speed-up, applying RaPTI

Table 5.3 shows how the same cases of invariance should be solved, using RaTI algorithm, that solves only one time-slice and gets the solution on all other slices by means of a simple change of variables. This is done on one processor and the time of computation is compared to the sequential time needed for solving the same slices.
The result is a tremendous, but not surprising, speed-up of the computations.

Case	1	2	3	4	5	6	7	8	9
p	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.4**	**0.3**	**0.2**	**0.1**
T	$_{4.9}10^{10}$	$_{2.4}10^{19}$	$_{1.8}10^{29}$	$_{2.8}10^{40}$	$_{1.5}10^{53}$	$_{5.9}10^{67}$	$_{4}10^{84}$	$_{1.4}10^{104}$	$_{3.5}10^{127}$
N	**1160**	**1179**	**1220**	**1306**	**1323**	**1491**	**1531**	**1304**	**1932**
$\mathbf{T_s}$	**4.62**	**4.70**	**5.26**	**5.42**	**5.55**	**6.25**	**6.68**	**5.62**	**8.25**
$\mathbf{T_1}$	**0.107**	**0.11**	**0.111**	**0.115**	**0.116**	**0.121**	**0.127**	**0.113**	**0.0.136**
$\mathbf{S_1}$	**43.22**	**42.69**	**47.36**	**47.11**	**47.87**	**51.68**	**52.60**	**49.77**	**60.69**

Table 5.3: Membrane - Cases of Invariance: Application of RaTI algorithm *by solving only one slice*, with $b = 1$ and $q = \frac{2p}{p+1}$.

5.4 Numerical Results: Case of Asymptotic Similarity

5.4.1 Existence of a Ratio Property

The asymptotic similarity of the rescaled systems, already proved, is a necessary condition for the existence of an asymptotic ratio property (i.e. $\lim_{n\to\infty} R_n = R_L$), as defined in proposition 8. However, this is not sufficient.
One needs also to have: $\lim_{n\to\infty} Z_n(s_n) = Z_L(s_L)$.

As detailed in theorem 1 and its corollary (given in Chapter 1), the latter convergence requires the following hypotheses to be satisfied:

1. the rescaled systems are asymptotically similar,
2. the sequence $\{s_n\}$ of rescaled times is uniformly upper bounded,
3. there exists $\hat{S} > 0$ such that: $\begin{cases} \forall s \in [0, s_L], & Z_L(s) \in B_{\hat{S}}, \\ \forall n, \ \ \forall s \in [0, s_n], & Z_n(s) \in B_{\hat{S}}, \end{cases}$
4. G_L verifies a Lipshitz condition on the domain $B_{\hat{S}}$,
5. there exists $\epsilon > 0$ such that H define bijections mapping an interval containing s_L and, for every n, an interval containing s_n, onto the interval $[-\epsilon, \epsilon] \subset \mathbb{R}$.

The asymptotic similarity is proved in the previous subsection and the existence of bijections is satisfied (see remark 3, in Chapter 2). The three other hypotheses are numerically verified in all the test cases, but they are not proved. This makes the asymptotic ratio property unproved, but numerically verified for all the tested cases, as shown below.

Table 5.4 gives a sample of the numerical ratio property obtained for some tested cases and gives also the ratio $R_L = \mathbf{1} + Z_L(s_L)$, corresponding to the limit problem of each case. Obviously, the ratios $\{R_n\}$ seem to become closer and closer to R_L, as n becomes larger.

Note that only the first component of the ratios is given in this table, since the second component follows directly from the first one, through the uniform EOS condition: $\forall n, \ \ 1 + Z_{n,2}(s_n) = |1 + Z_{n,1}(s_n)|^{\frac{p+1}{2}} \iff R_{n,2} = |R_{n,1}|^{\frac{p+1}{2}}$ and, similarly, $R_{L,2} = |R_{L,1}|^{\frac{p+1}{2}}$.

	Case 1	Case 2	Case 3	Case 4
	$p = 0.8$, $q = 0.84$	$p = 0.7$, $q = 0.77$	$p = 0.6$, $m = 0.66$	$p = 0.5$, $m = 0.55$
n	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,1}}$
1	32.01638228	35.14519119	31.96552779	31.80221657
2	18.28056309	19.17433786	13.88417528	12.0969595
3	12.4544877	12.69361351	8.678549884	7.336415221
⋮	⋮	⋮	⋮	⋮
331	1.06918957	1.069653695	1.039519338	1.033450754
332	1.068978459	1.06944921	1.039393128	1.033352919
333	1.068768635	1.069245963	1.03926773	1.033255658
⋮	⋮	⋮	⋮	⋮
476	1.047930393	1.049027608	1.026942978	1.023482069
477	1.047829064	1.048929175	1.026883374	1.023433977
478	1.047728163	1.048831157	1.026824022	1.023386083
⋮	⋮	⋮	⋮	⋮
1060	1.021414334	1.023274032	1.011325179	1.010698606
1061	1.021394043	1.023254365	1.011313186	1.010688666
1062	1.02137379	1.023234735	1.011301216	1.010678744
⋮	⋮	⋮	⋮	⋮
1486	1.01524735	1.017307495	1.007678937	1.007665909
1487	1.015237039	1.017297544	1.007672842	1.007660819
1488	1.015226741	1.017287606	1.007666755	1.007655735
⋮	⋮	⋮	⋮	⋮
1512	1.014983687	1.017053086	1.007523101	1.007535729
1513	1.014973727	1.017043478	1.007517215	1.007530811
1514	1.014963781	1.017033882	1.007511336	1.007525899
⋮	⋮	⋮	⋮	⋮
2999	1.007518218	1.009912732	1.003139833	1.003831921
3000	1.007515689	1.009910354	1.003138374	1.003830661
$\mathbf{R_{L,1}}$	**0.999874663**	**1.00462549**	**0.996705614**	**1.000071701**

Table 5.4: Membrane: Ratio Property in a Case of Asymptotic Similarity

5.4.2 Application of RaPTI

Tables (5.5) and (5.6) below summarize some results that were obtained by applying RaPTI algorithm on the membrane problem, in the case of asymptotic similarity when $p \leq q < \frac{2p}{p+1}$ and for 8 combinations of the problem parameters p and q, with $b = 1$.

When doing the preliminary analysis, the procedure FIND_RATIOS_MODEL, described in section 4.3.2 (of Chapter 4) and providing a mathematical model for predicting the ratios, could ultimately be repeated for every combination of the problem parameters (p and q), and for every change in the initial condition.
However, the numerical experiments show that:
- the common global behavior of the solution for the considered set of combinations of problem parameters $\left(p \leq q < \frac{2p}{p+1}\right)$,
- the common EOS condition and choice of $\{\beta_n\}$ used in the rescaling technique,
- and the resulting similarity and ratio properties,
make the same mathematical form of the model R_{fit}, with the same number n_{prec} of previous data to be used, hold the extrapolation test for all combinations of problem parameters and for any initial condition (within the same global behavior, of course), as well as at all iterations.
Hence, the procedure FIND_RATIOS_MODEL is implemented, once for all, in the preliminary stage, for an arbitrary combination (p, q) satisfying $p \leq q < \frac{2p}{p+1}$ and for an arbitrary initial condition.
The resulting mathematical model that has been used for the predictions of the ratios is a polynomial of degree $d = 2$, obtained by fitting the last $n_{prec} = 3$ computed ratios, in the least-squares sense.

The total number of slices N, and therefore the interval of integration $[0, T]$, corresponds to the maximum (or almost) number preventing the explosive solution from exceeding the machine capacity.

The total number of iterations vary from one case to another, but in all cases, the results show how small is this number compared to the total number of slices. This ascertains the fast convergence of RaPTI when applied to the membrane problem.

In all the experiments, the tolerance used for getting the time-step τ is $\epsilon_{tol}^{\tau} = 10^{-14}$. Similarly, the tolerance up to which is reached the EOS condition is $\epsilon_{tol}^{EOS} = 10^{-14}$.

The only differences between tables (5.5) and (5.6)are:
- the tolerance $\epsilon_{tol}^{n_s}$, used for getting n_s and
- the tolerance ϵ_{tol}^{g}, on the gaps.

Case	1	2	3	4	5	6	7	8
p	**0.8**	**0.7**	**0.7**	**0.6**	**0.6**	**0.6**	**0.5**	**0.5**
q	**0.84**	**0.74**	**0.77**	**0.66**	**0.69**	**0.72**	**0.55**	**0.60**
T	$_{1.0}10^{14}$	$_{1.6}10^{29}$	$_{2.7}10^{28}$	$_{8.3}10^{13}$	$_{1.1}10^{18}$	$_{2.4}10^{30}$	$_{2.5}10^{17}$	$_{1.3}10^{25}$
N	**65000**	**65000**	**50000**	**65000**	**65000**	**65000**	**65000**	**65000**
$\mathbf{T_s}$	**252.81**	**255.50**	**199.94**	**265.90**	**262.89**	**261.47**	**264.53**	**262.67**
n_s	1499	1143	1471	1156	1414	1993	1053	1385
n_I	6	11	12	35	28	23	5	5
T_2	134.81	132.27	103.37	135.87	135.66	136.64	135.28	135.17
$\mathbf{E_2}$	**0.94**	**0.97**	**0.97**	**0.98**	**0.97**	**0.96**	**0.98**	**0.97**
$\mathbf{S_2}$	**1.88**	**1.93**	**1.93**	**1.96**	**1.94**	**1.91**	**1.96**	**1.94**
S_2^{max}	1.95	1.97	1.94	1.97	1.96	1.94	1.97	1.96
T_4	70.91	69.74	57.11	74.06	73.77	76.06	71.95	72.34
$\mathbf{E_4}$	**0.89**	**0.92**	**0.88**	**0.90**	**0.89**	**0.86**	**0.92**	**0.91**
$\mathbf{S_4}$	**3.57**	**3.66**	**3.50**	**3.59**	**3.56**	**3.44**	**3.68**	**3.63**
S_4^{max}	3.74	3.80	3.68	3.80	3.75	3.66	3.81	3.76
T_8	39.07	37.82	32.10	40.47	41.18	43.25	38.80	39.88
$\mathbf{E_8}$	**0.81**	**0.84**	**0.78**	**0.82**	**0.80**	**0.76**	**0.85**	**0.82**
$\mathbf{S_8}$	**6.47**	**6.76**	**6.23**	**6.57**	**6.38**	**6.05**	**6.82**	**6.59**
$S8^{max}$	6.89	7.12	6.63	7.11	6.94	6.59	7.19	6.96

Table 5.5: Membrane Problem - Cases of Asymptotic Similarity: Application of RaPTI, when $\epsilon_{tol}^{g} = 5 \times 10^{-6}$, $\epsilon_{tol}^{n_s} = 10^{-5}$.

Actually, RaPTI has been tested on the previous 8 cases, using 2, 3, 4, 5, 6, 7, and 8 processors.
Figure 5.8 shows how the values of the efficiency and speed-up, averaged on the 8 cases of table 5.5, vary with the number of processors. It shows also how large is the efficiency and how close is the speed-up to the maximum speed-up stated by Amdhal's law.

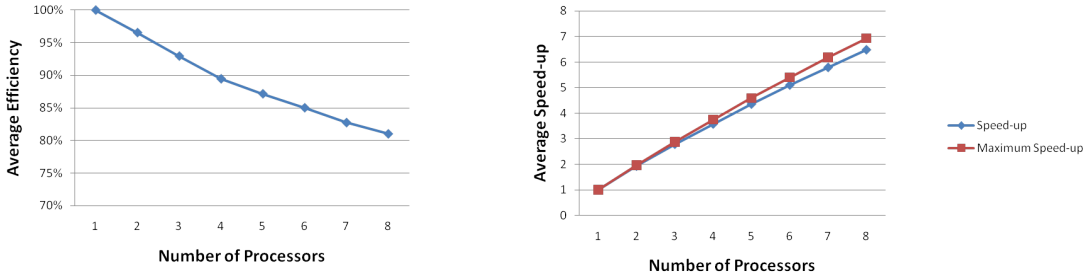
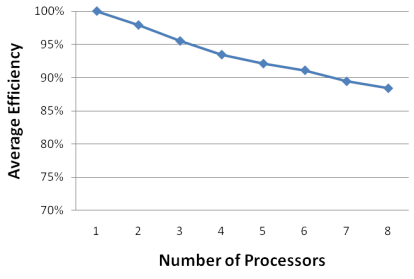


Figure 5.8: Membrane Problem - Cases of Asymptotic Similarity: Efficiencies and speed-ups, $\epsilon_{tol}^{g} = 5 \times 10^{-6}$, $\epsilon_{tol}^{n_s} = 10^{-5}$.

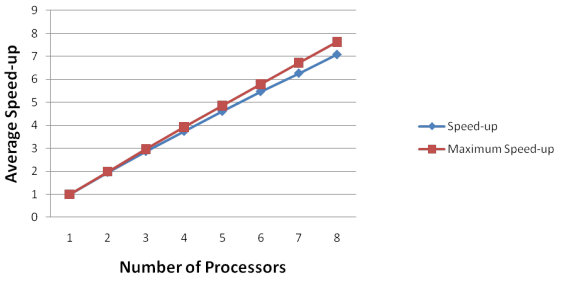
Case	1	2	3	4	5	6	7	8
p	**0.8**	**0.7**	**0.7**	**0.6**	**0.6**	**0.6**	**0.5**	**0.5**
q	**0.84**	**0.74**	**0.77**	**0.66**	**0.69**	**0.72**	**0.55**	**0.60**
T	$_{1.0}10^{14}$	$_{1.6}10^{29}$	$_{2.7}10^{28}$	$_{8.3}10^{13}$	$_{1.1}10^{18}$	$_{2.4}10^{30}$	$_{2.5}10^{17}$	$_{1.3}10^{25}$
N	**65000**	**65000**	**50000**	**65000**	**65000**	**65000**	**65000**	**65000**
$\mathbf{T_s}$	**252.81**	**255.50**	**199.94**	**265.90**	**262.89**	**261.47**	**264.53**	**262.67**
n_s	469	361	462	365	451	701	323	422
n_I	6	14	16	37	29	34	5	5
T_2	130.55	130.08	101.42	134.65	135.36	136.22	133.48	132.94
$\mathbf{E_2}$	**0.97**	**0.98**	**0.99**	**0.99**	**0.97**	**0.96**	**0.99**	**0.99**
$\mathbf{S_2}$	**1.94**	**1.96**	**1.97**	**1.97**	**1.94**	**1.92**	**1.98**	**1.98**
S_2^{max}	1.99	1.99	1.98	1.99	1.99	1.98	1.99	1.99
T_4	67.06	67.68	53.73	71.82	71.13	71.99	68.90	69.69
$\mathbf{E_4}$	**0.94**	**0.94**	**0.93**	**0.93**	**0.92**	**0.91**	**0.96**	**0.94**
$\mathbf{S_4}$	**3.77**	**3.78**	**3.72**	**3.70**	**3.70**	**3.63**	**3.84**	**3.77**
S_4^{max}	3.92	3.93	3.89	3.93	3.92	3.87	3.94	3.92
T_8	35.48	35.77	28.57	37.61	38.12	38.82	35.90	36.30
$\mathbf{E_8}$	**0.89**	**0.89**	**0.87**	**0.88**	**0.86**	**0.84**	**0.92**	**0.90**
$\mathbf{S_8}$	**7.13**	**7.14**	**7.00**	**7.07**	**6.90**	**6.74**	**7.37**	**7.24**
S_8^{max}	7.62	7.70	7.51	7.70	7.63	7.44	7.73	7.65

Table 5.6: Membrane Problem - Cases of Asymptotic Similarity: Application of RaPTI, when $\epsilon_{tol}^{g} = 10^{-5}$, $\epsilon_{tol}^{ns} = 10^{-4}$.

Here also, RaPTI has been tested on the previous 8 cases, using 2, 3, 4, 5, 6, 7, and 8 processors, and figure 5.9 shows how the values of the efficiency and speed-up, averaged on the 8 cases of table 5.6, vary with the number of processors. It shows also how large is the efficiency and how close is the speed-up to the maximum speed-up stated by Amdhal's law.



(a) Efficiency



(b) Efficiency

Figure 5.9: Membrane Problem - Cases of Asymptotic Similarity: Efficiencies and speed-ups, $\epsilon_{tol}^{g} = 10^{-5}$, $\epsilon_{tol}^{ns} = 10^{-4}$.

As expected:
- Larger tolerances imply better efficiencies and speed-ups,
- The efficiencies decrease, as the number of processors increases from 2 to 8, due to communication overhead,
- The speed-ups increase, as the number of processors increases.

Impact of the choices of ϵ_{tol}^{g} and $\epsilon_{tol}^{n_s}$:
Some additional experiments have been done, on 8 processors only, for analyzing the effect of a variation of the tolerance $\epsilon_{tol}^{n_s}$, or ϵ_{tol}^{g}, on the numbers n_s and n_I, on the efficiency, on the speed-up and on the accuracy of the solution.
For that purpose, a relative error has been computed, for comparing T_n^c and $Y_{n,i}^c$ to the sequentially computed end-of-slice values T_n and $Y_{n,i}$:

$$err_{rel} = \max_{1\leq n\leq N}\left[\max_{i\in\{1,2\}}\frac{\sqrt{\left(Y_{n,i}^c - Y_{n,i}\right)^2 + (T_n^c - T_n)^2}}{\sqrt{Y_{n,i}^2 + T_n^2}}\right],$$

Case	1	2	3	4	5	6	7	8
p	0.8	0.7	0.7	0.6	0.6	0.6	0.5	0.5
q	0.84	0.74	0.77	0.66	0.69	0.72	0.55	0.60
T	$1.0\,x10^{14}$	$1.6\,x10^{29}$	$2.6\,x10^{28}$	$8.3\,x10^{13}$	$1.1\,x10^{18}$	$2.4\,x10^{30}$	$2.5\,x10^{17}$	$1.3x10^{25}$
N	65000	65000	50000	65000	65000	65000	65000	65000
T_s	252.81	255.5	199.94	265.9	262.89	261.47	264.53	262.67

Choice I	n_s	1499	1143	1471	1156	1414	1993	1053	1385
	n_I	6	11	12	35	28	23	5	5
	err_{rel}	1.9E-02	2.5E-02	1.1E-02	8.2E-03	1.3E-02	8.2E-03	2.1E-02	2.6E-02
$\varepsilon_{tol}^{n_s} = 10^{-5}$	T_8	39.07	37.82	32.1	40.47	41.18	43.25	38.8	39.88
	E_8	0.809	0.844	0.779	0.821	0.798	0.756	0.852	0.823
$\varepsilon_{tol}^{g} = 5\times10^{-6}$	S_8	6.471	6.756	6.229	6.570	6.384	6.046	6.818	6.587
	S_8^{max}	6.888	7.123	6.634	7.114	6.943	6.586	7.185	6.962

Choice II	n_s	1499	1143	1471	1156	1414	1993	1053	1385
	n_I	5	9	9	32	24	18	4	4
$\varepsilon_{tol}^{n_s} = 10^{-5}$	err_{rel}	5.3E-02	5.9E-02	8.1E-02	1.2E-02	1.9E-02	3.7E-02	6.1E-02	7.8E-02
	T_8	40.08	39.27	32.69	41.9	42.5	44.44	39.71	40.96
$\varepsilon_{tol}^{g} = 10^{-5}$	E_8	0.788	0.813	0.765	0.793	0.773	0.735	0.833	0.802
	S_8	6.308	6.506	6.116	6.346	6.186	5.884	6.662	6.413
	S_8^{max}	6.888	7.123	6.634	7.114	6.943	6.586	7.185	6.962

Choice III	n_s	469	361	462	365	451	701	323	422
	n_I	6	14	16	37	29	34	5	5
$\varepsilon_{tol}^{n_s} = 10^{-4}$	err_{rel}	4.5E-02	3.1E-02	4.4E-02	1.2E-02	2.2E-02	2.5E-02	6.8E-02	9.1E-02
	T_8	35.48	35.77	28.57	37.61	38.12	38.82	35.9	36.3
$\varepsilon_{tol}^{g} = 10^{-5}$	E_8	0.891	0.893	0.875	0.884	0.862	0.842	0.921	0.905
	S_8	7.125	7.143	6.998	7.070	6.896	6.735	7.369	7.236
	S_8^{max}	7.615	7.701	7.514	7.697	7.629	7.438	7.731	7.652

Figure 5.10: Membrane - Impact of the choices of ϵ_{tol}^{g} and $\epsilon_{tol}^{n_s}$.

The impact of ϵ_{tol}^{g} can be deduced by comparing the choices I and II of tolerances, given in the table of figure 5.10.
For a given $\epsilon_{tol}^{n_s} = 10^{-5}$, the number n_s and the maximum speed-up are unchanged, whereas decreasing the tolerance on the gaps (choice I) makes:

- the number of iterations increases (by almost 22%, in average),
- the error decreases (by almost 60%, in average),
- the efficiency and the speed-up increase (by almost 2.8%, in average), in spite of the increasing n_I.

Figure 5.11 compares the resulting speed-ups.

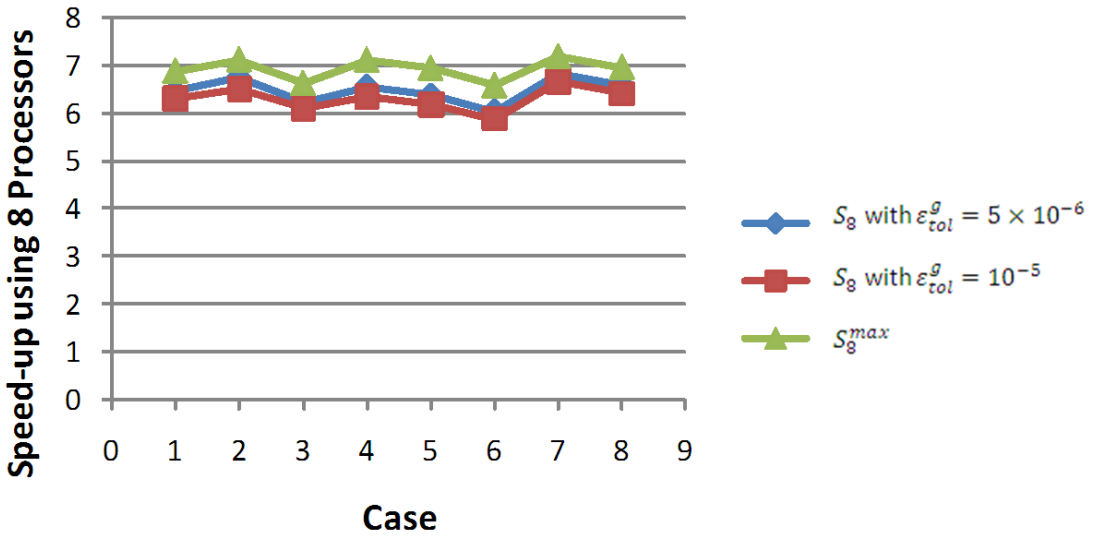


Figure 5.11: Membrane - Comparative Speed-ups, for different ϵ_{tol}^g.

The impact of $\epsilon_{tol}^{n_s}$ is noticed by comparing the choices II and III of tolerances given in the table of figure 5.10.
For a given $\epsilon_{tol}^g = 10^{-5}$, increasing the tolerance on n_s (choice III) makes:

- the number n_s decreases significantly (by almost 68%, in average),
- the number of iterations increases (by almost 41%, in average),
- the relative error sometimes increases and some other times decreases,
- the efficiency increases (by almost 12%, in average),
- the speed-up increases (by almost 12%, in average) and the maximum speed-up increases (by almost 10%, in average) .

Figure 5.12 compares the resulting speed-ups.

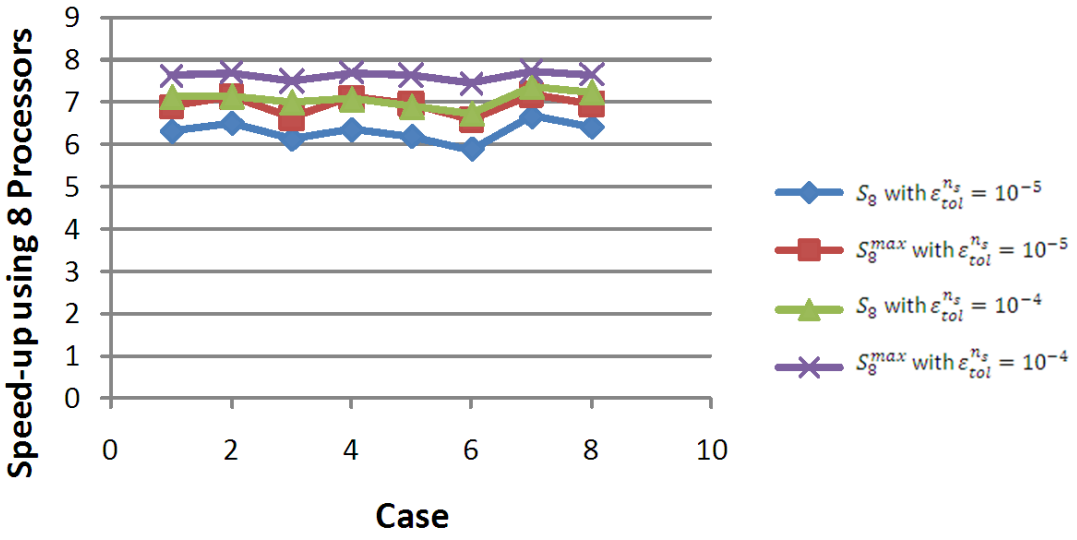


Figure 5.12: Membrane - Comparative Speed-ups, for different $\epsilon_{tol}^{n_s}$.

From this comparative study, and since the goal is to have a good speed-up and a good accuracy, one deduces that:

- The accuracy is almost not affected by any change of $\epsilon_{tol}^{n_s}$, but improves a lot with smaller ϵ_{tol}^{g}.
- The speed-up is almost not affected by any change of ϵ_{tol}^{g}, but improves quite well with larger $\epsilon_{tol}^{n_s}$.

Thus, one should optimize the choices of the tolerances by decreasing ϵ_{tol}^{g} and increasing $\epsilon_{tol}^{n_s}$, as much as possible.

Chapter 6

Reaction-Diffusion Problem

Consider the reaction-diffusion problem of the form:

$$\frac{\partial u}{\partial t} - \Delta u^m = a u^p,\ x \in \Omega \subset \mathbb{R}^d, \quad u(x,t) = 0,\ x \in \partial\Omega \quad u(x,0) = u_0(x) > 0,$$

where $a > 0$, $m, p > 0$.
After a space semi-discretization of dimension K, it reduces to the general form (S) of a first order initial value problem, of dimension K.

When the problem parameters m and p are such that $0 < m \leq p \leq 1$, the solution exhibits, on $[0, \infty)$, an explosive behavior.
The application of our time-slicing and rescaling techniques yields an asymptotic ratio property, allowing a quick convergence of RaPTI algorithm and a good speed-up (evaluated as described in Appendix 2).

The particular case $m = p = 1$, simplifying the problem to a linear one (similar to that discussed in chapter 2), is independently tested and the application of RaPTI yields a very good speed-up.

The notations used in this chapter are those indicated in table 10.1 and those given in the following definition.

Definition 9 • *$V.^\alpha$ is the component-wise exponentiation to the power α:*

$$\forall \alpha \in \mathbb{R}, \quad \forall V = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{pmatrix} \in \mathbb{R}^K \text{ with } \forall i,\ V_i > 0, \qquad V.^\alpha = \begin{pmatrix} V_1^\alpha \\ V_2^\alpha \\ \vdots \\ V_K^\alpha \end{pmatrix} \tag{6.1}$$

- *D_V is the diagonal $K \times K$ matrix, having the vector V on its main diagonal:*

$$\forall V = \begin{pmatrix} V_1 \\ V_2 \\ \vdots \\ V_K \end{pmatrix} \in \mathbb{R}^K, \qquad D_V = \begin{pmatrix} V_1 & 0 & \cdots & 0 \\ 0 & V_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V_K \end{pmatrix} \in \mathbb{R}^{K \times K} \tag{6.2}$$

6.1 Description of the problem

Consider the reaction-diffusion problem defined by:

$$\begin{cases} \frac{\partial u}{\partial t} - \Delta u^m = au^p,\ x \in \Omega \subset \mathbb{R}^d,\ t > 0 & (6.3.1) \\ u(x,t) = 0,\ x \in \partial\Omega,\ t \geq 0, & (6.3.2) \\ u(x,0) = u_0(x) > 0,\ x \in \Omega. & (6.3.3) \end{cases} \tag{6.3}$$

where $a > 0$, $m > 0$, $p > 0$, and Δ is the Laplace operator.
u is a concentration which verifies a positivity property, see [61], Δu^m is the diffusion term and au^p is the reaction term that determines the behavior of the solution.
Note that as m, p and the initial conditions vary, the behavior of the solution varies as well, as given in [61] and summarized below.

p,m	**Behavior of the Solution**
$0 < p < m$	Bounded behavior and global time existence
$0 < m \leq p \leq 1$	Global time existence and explosive behavior for certain initial conditions
$0 < m \leq 1 < p$	Finite time existence

Table 6.1: RD Problem - Different Behaviors of the Solution

In the present Chapter, we are interested in the case where:

$$0 < m \leq p \leq 1, \tag{6.4}$$

whose solution can, for some initial conditions, go monotonously to infinity in infinite time (otherwise, it vanishes in finite time), as proved in [61].

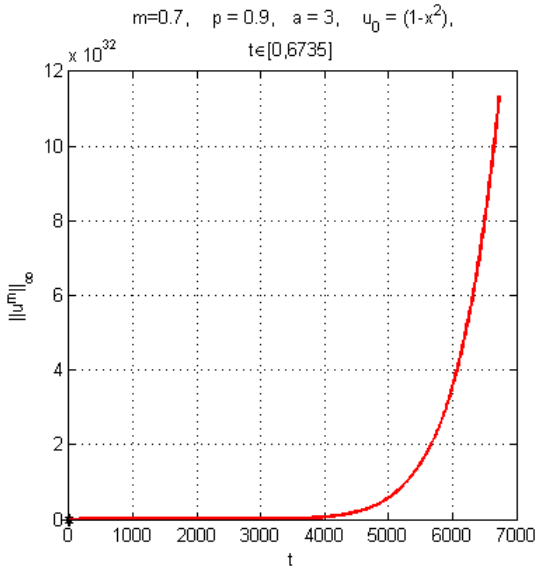


Figure 6.1: Reaction-Diffusion Problem - Global Behavior, when $m = 0.7$, $p = 0.9$, $a = 3$ and $u_0(x) = 1 - x^2$, with $\Omega = [-1, 1] \subset \mathbb{R}$.

The particular case $m = p = 1$, simplifies the problem to a linear one, similar to that discussed in chapter 2, and is numerically tested in the last section of the chapter.

The first step, for solving problem (6.3), is to introduce a preliminary change of variable: $v = u^m$. By letting also $q = \frac{1}{m}$, the problem translates to:

$$\begin{cases} \frac{\partial v}{\partial t} = \frac{1}{q}\frac{1}{v^{q-1}}\Delta v + \frac{a}{q}v^{pq-q+1},\ x \in \Omega \subset \mathbb{R}^d,\ t > 0 & (6.5.1) \\ v(x,t) = 0,\ x \in \partial\Omega,\ t \geq 0, & (6.5.2) \\ v(x,0) = v_0(x) > 0,\ x \in \Omega. & (6.5.3) \end{cases} \tag{6.5}$$

with $v_0(x) = [u_0(x)]^m = [u_0(x)]^{\frac{1}{q}}$, and $0 < \frac{1}{q} \leq p \leq 1$, yielding $q \geq 1$ and $2 - q \leq pq - q + 1 \leq 1$.

Next, a space semi-discretization of dimension K is applied to the interior of the domain Ω, using the finite-difference operator A in place of the Laplacian Δ, where $A \in \mathbb{R}^{K\times K}$ is a sparse symmetric positive definite matrix that discretizes the operator $-\Delta$.
The solution $v(x,t)$ and its initial value $v_0(x)$ are then approximated by the vectors:

$$v(x,t) \approx Y(t) = \begin{pmatrix} Y_1(t) \\ Y_2(t) \\ \vdots \\ Y_K(t) \end{pmatrix} \quad \text{and} \quad v_0(x) \approx Y_0 = \begin{pmatrix} Y_{0,1} \\ Y_{0,2} \\ \vdots \\ Y_{0,K} \end{pmatrix}.$$

Problem (6.5) is now equivalent to the initial value problem:

$$\begin{cases} \frac{dY}{dt} = F_{diff}(Y) + F_{reac}(Y), \quad t > 0 & (6.6.1) \\ Y(0) = Y_0, & (6.6.2) \end{cases} \tag{6.6}$$

where $F_{diff}(Y)$ and $F_{reac}(Y)$ correspond respectively to the diffusion and reaction terms of the problem:

$$F_{diff}(Y) = -\frac{1}{q}D^{-1}_{Y^{q-1}}AY, \tag{6.7}$$

$$F_{reac}(Y) = \frac{a}{q}Y^{pq-q+1}. \tag{6.8}$$

Note that problem (6.6) has the form of the general initial value problem (S), with:

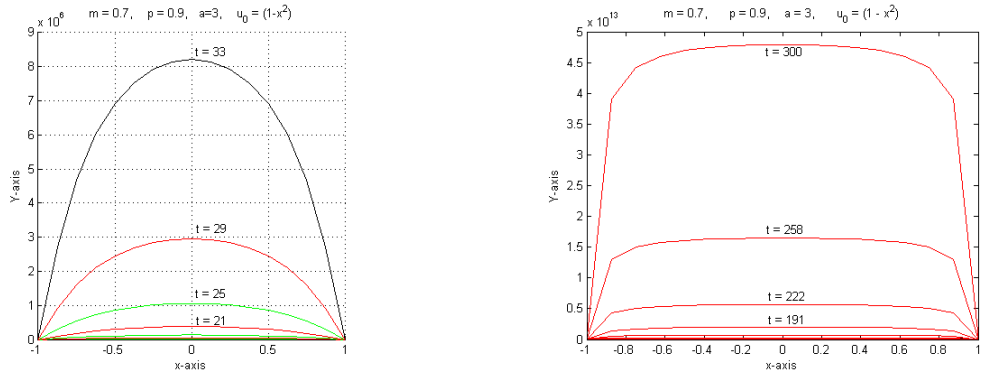
$$F(Y) = F_{diff}(Y) + F_{reac}(Y). \tag{6.9}$$

We did not find (at this point) any reference specifying where is the blow-up occurring and how each component of the solution varies. Our numerical integration of (6.3), for cases satisfying (6.4) motivates the following assumptions.

We consider, in this chapter, that the behavior of the solution satisfies the two following assumptions:

Assumption 3 *Each component of the solution monotonously increases toward ∞.*

Assumption 4 *The blow-up occurs on all the components of the solution.*



(a) $0 \leq t \leq 33$ (b) $0 \leq t \leq 300$

Figure 6.2: Reaction-Diffusion Problem - Solution Profile, when $m = 0.7$, $p = 0.9$, $a = 3$ and $u_0(x) = 1 - x^2$, with $\Omega = [-1, 1] \subset \mathbb{R}$.

6.2 Rescaling and Similarity Properties

6.2.1 Choice of an EOS condition

Since the reaction-diffusion problem (6.3) is known to have a positive solution u, it follows that the solution Y of the initial value problem (6.6) is positive in all its components: $\forall t > 0,\ \forall i,\ \ Y_i(t) > 0$ As discussed in Chapter 2, in case of explosive behavior of the solution (and under the nonzeroness condition), the functions $\{E_n\}$ governing the EOS condition could be that defined in (2.6), for all $n > 0$, by:

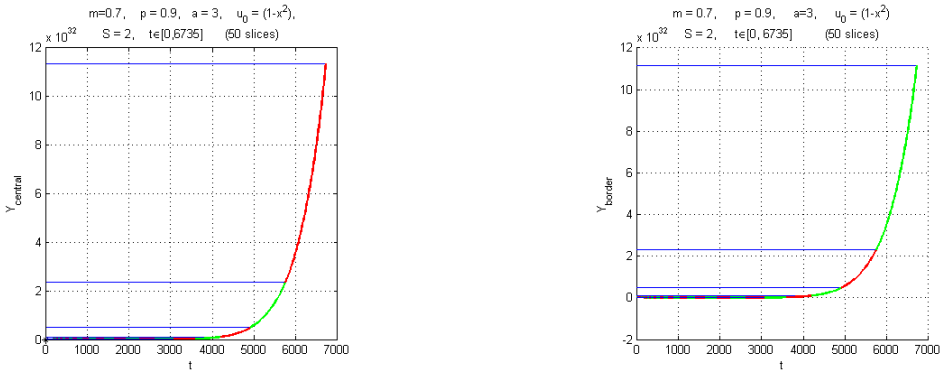
$$E_n(W) = \left\| D_n^{-1} \left(W - Y_{n-1} \right) \right\|_\infty - S.$$

For a given $S > 0$, the n^{th} slice is then ended as soon as the solution satisfies:

$$\begin{array}{c} \text{at } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0 \iff \left\| D_n^{-1} \left(Y(T_n) - Y_{n-1} \right) \right\|_\infty = S, \\ \text{with} \quad \left\| D_n^{-1} \left(Y(t) - Y_{n-1} \right) \right\|_\infty < S, \quad \text{if } \ T_{n-1} < t < T_n, \end{array} \tag{6.10}$$

thus, preventing the absolute value of the relative variation of each component of $Y(t)$ relatively to the same component of Y_{n-1} from exceeding a cut-off value S.
The explosive behavior of the solution makes such EOS condition guaranteed to be reached. Moreover, if $||Y(.)||_\infty$ monotonously increases with t, one has:
$\forall t > T_n,\ \ \left\| D_n^{-1} \left(Y(t) - Y_{n-1} \right) \right\|_\infty > S.$

Figure 6.3 represents the central and border component of the solution Y as a function of t, on the first 50 slices and for the same case than that represented in figures (6.1) and (6.2).



(a) Central Component (b) Border Component

Figure 6.3: Reaction-Diffusion Problem - EOS Condition, when $m = 0.7$, $p = 0.9$, $a = 3$ and $u_0(x) = 1 - x^2$, with $\Omega = [-1, 1] \subset \mathbb{R}$.

Problem (6.6) is then equivalent to the sequence of initial value shooting problems in which one seeks, on each n^{th} slice ($n \geq 1$), for the time T_n and the solution $Y : [T_{n-1}, T_n] \longrightarrow \mathbb{R}^K$ such that:

$$\begin{cases} \frac{dY}{dt} = F_{diff}(Y) + F_{reac}(Y), & (6.11.1) \\ Y(T_{n-1}) = Y_{n-1}, & (6.11.2) \\ E_n\left[Y(T_n)\right] = 0 \text{ and } \forall t \in (T_{n-1}, T_n), \;\; E_n\left[Y(t)\right] \neq 0, & (6.11.3) \end{cases} \tag{6.11}$$

where $F_{diff}(Y)$ and $F_{reac}(Y)$ are given in (6.7) and (6.8).

6.2.2 Rescaling

Change of Variables:
Note first that the positivity of Y makes the nonzeroness condition (1.11) hold, for the end-of-slice values, and problem (6.11) can be rescaled through the change of variables (1.12) given by:

$$\begin{cases} t = T_{n-1} + \beta_n s, \qquad \beta_n > 0 & (6.12.1) \\ Y(t) = D_n\left(\mathbf{1} + Z_n(s)\right), & (6.12.2) \end{cases} \tag{6.12}$$

where $D_n = D_{Y_{n-1}}$, due to the nonzeroness condition.

Rescaled EOS Condition:
As shown in (2.8), the functions $\{E_n\}$ governing the EOS condition rescale to a unique function $H : \mathbb{R}^K \to \mathbb{R}$ defined, independently of n, by:

$$\forall W \in \mathbb{R}^K, \;\; H(W) = \|W\|_\infty - S.$$

For a given cutoff value $S > 0$, the n^{th} slice is then ended as soon as the rescaled solution $Z(.)$ satisfies the rescaled EOS condition:

$$\begin{array}{lll} \text{at } s = s_n, & H\left[Z_n(s_n)\right] = 0 \iff & \|Z_n(s_n)\|_\infty = S, \\ \text{with} & \|Z_n(s)\|_\infty < S, & \text{if } 0 < s < s_n. \end{array} \tag{6.13}$$

Rescaled Systems:
It follows from (6.11) and (6.12) that:

$$\begin{aligned}\frac{dZ_n}{ds} &= \beta_n D_n^{-1} F\big(D_n\left[\mathbf{1}+Z_n(s)\right]\big)\\ &= \beta_n D_n^{-1} F_{diff}\big(D_n\left[\mathbf{1}+Z_n(s)\right]\big) + \beta_n D_n^{-1} F_{reac}\left(D_n\left[\mathbf{1}+Z_n(s)\right]\right)\end{aligned}.$$

The resulting rescaled systems are:

$$\begin{cases} \frac{dZ_n}{ds} = G_n(Z_n) \quad 0 < s \leq s_n, & (6.14.1)\\ Z_n(0) = 0, & (6.14.2)\\ \forall s < s_n,\ \|Z_n(s)\|_\infty < S,\ \text{and}\ \|Z_n(s_n)\|_\infty = S. & (6.14.3)\end{cases} \tag{6.14}$$

where:

$$G_n(Z_n) = G_{n_{diff}}(Z_n) + G_{n_{reac}}(Z_n) \tag{6.15}$$

with the following expressions for the diffusion and reaction terms of G_n:

$$G_{n_{diff}}(Z_n) = -\frac{1}{q}\beta_n D^{-1}_{(Y_{n-1}).^q} D^{-1}_{[\mathbf{1}+Z_n(s)].^{q-1}} A D_{Y_{n-1}}\left[\mathbf{1}+Z_n(s)\right], \tag{6.16}$$

$$G_{n_{reac}}(Z_n) = \frac{a}{q}\beta_n D_{(Y_{n-1}).^{pq-q}}\left[\mathbf{1}+Z_n(s)\right].^{pq-q+1}. \tag{6.17}$$

Positivity and Boundedness of Z_n:
Under assumption (3), one deduces: $\forall t,\ \forall i,\ Y_i(t) > Y_{n-1,i}$.
Since $\forall n,\ \forall t \in [T_{n-1}, T_n],\ \forall i,\ Y_i(t) = Y_{n-1,i}\left(1 + Z_{n,i}(s)\right)$, it follows the positivity of Z_n on all slices:

$$\forall n,\ \forall s \in [0, s_n],\ \forall i,\ \ Z_{n,i}(s) \geq 0, \tag{6.18}$$

Together with EOS condition (6.13), (6.18) yields:

$$\forall n,\ \forall s \in [0, s_n],\ \forall i,\ \ 0 \leq Z_{n,i}(s) \leq S, \tag{6.19}$$

Let for all $S > 0$,

$$B_S^+ = \{V \in \mathbb{R}^K,\ \forall i,\ 0 \leq V_i \leq S\}. \tag{6.20}$$

(6.19) is then equivalent to:

$$\forall n,\ \forall s \in [0, s_n],\ Z_n(s) \in B_S^+. \tag{6.21}$$

Some Useful Signs:
Note that $0 < m = \frac{1}{q} \leq p \leq 1$, implies:

$$\begin{cases} q - 1 > 0,\\ pq \geq 1,\\ -q < 1 - q \leq pq - q \leq 0,\\ 2 - q \leq pq - q + 1 \leq 1 \end{cases} \tag{6.22}$$

with:

$$\begin{cases} 2 - q > 0 & \text{if } 1/2 < m \leq 1\\ 2 - q = 0 & \text{if } m = 1/2\\ 2 - q > 0 & \text{if } 0 < m < 1/2 \end{cases} \tag{6.23}$$

6.2.3 Critical Choice of $\{\beta_n\}$

Remark 8 ***: Choice yielding a uniform similarity***
For solving sequentially such explosive problem, one would aim at reaching a uniform similarity of the rescaled systems, in order to control the stiffness of the problem.
Let $A = (a_{ij})$. The i^{th} component of the expressions (6.7) and (6.8) for $F_{diff}(Y)$ and

$F_{reac}(Y)$ expresses as:

$$[F_{diff}(Y)]_i = -\frac{1}{q}\frac{1}{Y_i^{q-1}}\sum_{j=1}^{K} a_{ij}Y_j = \sum_{j=1}^{K} -\frac{1}{q}a_{ij}Y_i^{1-q}Y_j$$

$$[F_{reac}(Y)]_i = \frac{a}{q}Y_i^{pq-q+1}.$$

One notices then, that $F_{diff}(.)$ and $F_{reac}(.)$ have the algebraic form of F given in (3.1), in Chapter 3, but the condition (3.2), assuming the powers are integers larger than 1, does not hold in this case. However, the positivity of Y, in the present problem, makes the analysis that has been done, in Chapter 3, hold for any real powers.
It follows a critical value of β_n, given in (3.9), that guarantees the uniform similarity of the rescaled systems and translates here to:

$$\beta_n = \frac{1}{\max\left[\max_{i,j}\left\{|a_{ij}|\,|Y_{n-1,i}|^{-q}\,|Yn-1,j|\right\}\ ,\ \max_i\left\{\frac{a}{q}Y_{n-1,i}^{pq-q}\right\}\right]}. \tag{6.24}$$

In the present chapter, our goal is to solve the problem in parallel and, for that purpose, we aim at reaching an asymptotic similarity, in order to get a strong ratio property. But the critical value (6.24) of β_n does not allow to define a limit problem and we rather need a symbolic expression that is likely to do the job.
Besides, it is well known that the pure diffusion problem vanishes with time [61]. It follows that the explosive behavior of the solution is due to the reaction term.
Thus, we will try to choose a time rescaling factor β_n, given explicitly in terms of Y_{n-1}, that could control the explosive reaction term and be likely to yield an asymptotic similarity.

Magnitude of $G_{n_{reac}}(Z_n)$:
The expression (6.17) of the reaction term of $G_n(.)$ yields:

$$\|G_{n_{reac}}(Z_n(s))\|_\infty \leq \frac{a}{q}\beta_n \left\|D_{(Y_{n-1}).^{pq-q}}\right\|_\infty \left\|[\mathbf{1}+Z_n(s)].^{pq-q+1}\right\|_\infty.$$

From (6.19) one deduces:

$$\forall n,\ \ \forall s\in[0,s_n],\quad 1\leq 1+Z_{n,i}(s)\leq 1+S,$$

implying for all i:

$$\min\left[1,(1+S)^{pq-q+1}\right] \leq [1+Z_{n,i}(s)]^{pq-q+1} \leq \max\left[1,(1+S)^{pq-q+1}\right],$$

depending on the sign of $pq-q+1$ deduced from (6.22) and (6.23). Therefore:

$$\left\|[\mathbf{1}+Z_n(s)].^{pq-q+1}\right\|_\infty \leq \max\left[1,(1+S)^{pq-q+1}\right]$$

Besides: $\left\|D_{(Y_{n-1}).^{pq-q}}\right\|_\infty = \|(Y_{n-1}).^{pq-q}\|_\infty$.
It follows:

$$\|G_{n_{reac}}(Z_n(s))\|_\infty \leq \frac{a}{q}\beta_n \left\|(Y_{n-1}).^{pq-q}\right\|_\infty \max\left[1,(1+S)^{pq-q+1}\right]. \tag{6.25}$$

Critical choice of β_n:
By choosing, for example, the "critical value":

$$\beta_n = \frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty}, \tag{6.26}$$

one gets the growth of $G_{n_{reac}}$ being controlled, since (6.25) and (6.26) yield:

$$\|G_{n_{reac}}(Z_n(s))\|_\infty \leq \frac{a}{q}\max\left[1,(1+S)^{pq-q+1}\right] = C_1. \tag{6.27}$$

For the critical choice (6.26) for $\{\beta_n\}$, the expressions of the reaction and diffusion terms of G_n are then:

$$G_{n_{diff}}(Z_n(s)) = \; - \; \frac{1}{q}\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty} D^{-1}_{(Y_{n-1}).^q} D^{-1}_{[\mathbf{1}+Z_n(s)].^{q-1}} A D_{Y_{n-1}}\left[\mathbf{1}+Z_n(s)\right] \tag{6.28}$$

$$G_{n_{reac}}(Z_n(s)) = \frac{a}{q}\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty} D_{(Y_{n-1}).^{pq-q}}\left[\mathbf{1}+Z_n(s)\right].^{pq-q+1} \tag{6.29}$$

6.2.4 Similarity Properties

Lemma 2
Under assumptions 3 and 4 and for the choice (6.26) for β_n, if $0 < m < p \leq 1$, then:

$$\forall W \in B_S^+, \quad \lim_{n\to\infty} \left\|G_{n_{diff}}(W)\right\|_\infty = 0, \tag{6.30}$$

where B_S^+ is defined in (6.20).

Proof: The expression (6.16) of the diffusion term of $G_n(.)$ yields:

$$\left\|G_{n_{diff}}(W)\right\|_\infty \leq \; \frac{1}{q}\beta_n \left\|D^{-1}_{(Y_{n-1}).^q}\right\|_\infty \left\|D^{-1}_{[\mathbf{1}+W].^{q-1}}\right\|_\infty \|A\|_\infty \left\|D_{Y_{n-1}}\right\|_\infty \|\mathbf{1}+W\|_\infty$$

with $W \in B_S^+$, implying:
$1 \leq \|\mathbf{1}+W\|_\infty \leq 1+S$ and $\left\|D^{-1}_{[\mathbf{1}+W].^{q-1}}\right\|_\infty \leq 1$ (since $q-1>0$).
Besides: $\left\|D_{Y_{n-1}}\right\|_\infty = \|Y_{n-1}\|_\infty$ and $\left\|D^{-1}_{(Y_{n-1}).^q}\right\|_\infty = \|(Y_{n-1}).^{-q}\|_\infty$.
Together with the choice (6.26) for β_n, this yields:

$$\left\|G_{n_{diff}}(W)\right\|_\infty \leq \; \frac{(1+S)\|A\|_\infty}{q}\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty}\left\|(Y_{n-1}).^{-q}\right\|_\infty \|Y_{n-1}\|_\infty .$$

On the other hand, one deduces from (6.1) that:

$$\forall V \in B_S^+, \;\; \forall \alpha > 0, \qquad \left\{ \begin{array}{l} \|V.^\alpha\|_\infty = \|V\|_\infty^\alpha , \\ \|V.^{-\alpha}\|_\infty = \left\|V.^{-1}\right\|_\infty^\alpha . \end{array} \right.$$

Since $pq-q<0$ and $-q<0$, it follows:

$$\left\|G_{n_{diff}}(W)\right\|_\infty \leq \; \frac{(1+S)\|A\|_\infty}{q}\frac{1}{\|(Y_{n-1}).^{-1}\|_\infty^{q-pq}}\left\|(Y_{n-1}).^{-1}\right\|_\infty^q \|Y_{n-1}\|_\infty . \tag{6.31}$$

But, by assumption 4, the explosion occurs on all components.
One gets then: $\forall i,\ \lim_{n\to\infty} |Y_{n-1,i}| = \|Y_{n-1}\|_\infty$, implying:
$\lim_{n\to\infty} \left\|(Y_{n-1}).^{-1}\right\|_\infty = \lim_{n\to\infty} \frac{1}{\|(Y_{n-1})\|_\infty}$.
Therefore:

$$\lim_{n\to\infty} \frac{1}{\left\|(Y_{n-1}).^{-1}\right\|_\infty^{q-pq}} \left\|(Y_{n-1}).^{-1}\right\|_\infty^{q} \|Y_{n-1}\|_\infty = \lim_{n\to\infty} \|Y_{n-1}\|_\infty^{1-pq}.$$

Knowing also that the explosive behavior of the solution yields $\lim_{n\to\infty} \|Y_{n-1}\| = \infty$ and that $0 < m < p \leq 1$ makes $1 - pq < 0$, one deduces then, from (6.31), that the diffusion term vanishes at infinity. ■

Theorem 8
Under the assumptions of lemma 2, the rescaled systems (6.14), together with the choice (6.26) for β_n, are asymptotically similar to the limit system:

$$\begin{cases} \frac{dZ_L}{ds} = G_L(Z_L), \quad 0 < s \leq s_L, & (6.32.1) \\ Z_L(0) = 0, & (6.32.2) \\ \forall s < s_L,\ \|Z_L(s)\|_\infty < S,\ \textit{and}\ \|Z_L(s_L)\|_\infty = S. & (6.32.3) \end{cases} \tag{6.32}$$

defined by the function:

$$G_L(Z_L) = \frac{a}{q}\left[\mathbf{1} + Z_L(s)\right].^{pq-q+1}. \tag{6.33}$$

Proof:
$\forall W \in B_S^+,\ G_n(W) = G_{n_{diff}}(W) + G_{n_{reac}}(W)$, with:
$\begin{cases} G_{n_{diff}}(W) = -\frac{1}{q}\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty} D^{-1}_{(Y_{n-1}).^q} D^{-1}_{(\mathbf{1}+W).^{q-1}} A D_{Y_{n-1}}(\mathbf{1}+W), \\ G_{n_{reac}}(W) = \frac{a}{q}\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty} D_{(Y_{n-1}).^{pq-q}}(\mathbf{1}+W).^{pq-q+1}. \end{cases}$
Therefore:
$\|G_n(W) - G_L(W)\|_\infty =$

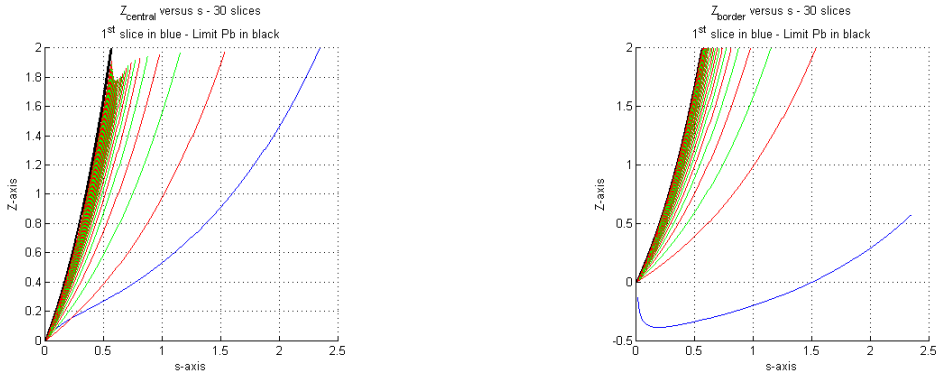
$$\left\| G_{n_{diff}}(W) + \left(\frac{a}{q}\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty} D_{(Y_{n-1}).^{pq-q}} - \frac{a}{q} I\right)(\mathbf{1}+W).^{pq-q+1} \right\|_\infty,$$

where I is the identity matrix in $\mathbb{R}^{K\times K}$.
Besides, it has been proved in lemma 2 that: $\forall W \in B_S^+$, $\lim_{n\to\infty} \left\|G_{n_{diff}}(W)\right\|_\infty = 0$.
Moreover, one has by assumption 4: $\forall i,\ \lim_{n\to\infty} \left(\frac{Y_{n-1,i}}{\|Y_{n-1}\|_\infty}\right) = 1$.
This implies: $\lim_{n\to\infty} \left(\frac{1}{\|(Y_{n-1}).^{pq-q}\|_\infty} D_{(Y_{n-1}).^{pq-q}}\right) = I$.
It follows:

$$\forall \rho \in [0, S],\ \forall W \in B_\rho^+,\ \lim_{n\to\infty} \left[\max_{W\in B_\rho^+} \|G_n(W) - G_L(W)\|_\infty\right] = 0.$$

Thus, the asymptotic similarity of the rescaled systems, according to definition 3 of Chapter 1 and restricted to B_S^+. ■

Figure 6.4 illustrates a case of asymptotic similarity: the rescaled solution $Z_n(s)$ on successive time-slices are represented on the same graph (central and border component only). Each curve, corresponding to the n^{th} slice $[0, s_n]$, starts at $Z_{n,i}(0) = 0$ and ends at $Z_{n,i}(s_n)$. One can notice how the successive curves become closer and closer to that corresponding to the limit problem (in black), as n becomes larger.



(a) Central Component (b) Border Component

Figure 6.4: Non Linear RD Problem - Z versus s, on 30 slices, when $m = 0.7$, $p = 0.9$, $a = 3$, $S = 2$, with $\Omega = [-1, 1] \subset \mathbb{R}$ and $u_0(x) = 1 - x^2$

Remark 9 *The case where $p = m = 1$ (implying $1 - pq = 0$) corresponds to the linear case of the reaction diffusion problem and presents also an asymptotic similarity as that described in Chapter 3.*

6.3 Numerical Results: Nonlinear case

6.3.1 Existence of a Ratio Property

The asymptotic similarity of the rescaled systems is a necessary condition for the existence of an asymptotic ratio property (i.e. $\lim_{n\to\infty} R_n = R_L$), as defined in proposition 8. However, this is not sufficient.
One needs also to have: $\lim_{n\to\infty} Z_n(s_n) = Z_L(s_L)$.

As detailed in theorem 1 and its corollary, the latter convergence requires the following hypotheses to be satisfied:

1. the rescaled systems are asymptotically similar,
2. the sequence $\{s_n\}$ of rescaled times is uniformly upper bounded,
3. there exists $\hat{S} > 0$ such that: $\begin{cases} \forall s \in [0, s_L], & Z_L(s) \in B_{\hat{S}}, \\ \forall n, \quad \forall s \in [0, s_n], & Z_n(s) \in B_{\hat{S}}, \end{cases}$
4. G_L verifies a Lipshitz condition on the domain $B_{\hat{S}}$,
5. there exists $\epsilon > 0$ such that H define bijections mapping an interval containing s_L and, for every n, an interval containing s_n, onto the interval $[-\epsilon, \epsilon] \subset \mathbb{R}$.

The asymptotic similarity is proved in the previous subsection, the existence of $\hat{S}$ results directly from the EOS condition and the existence of bijections is satisfied (see remark 2, in Chapter 2). The two other hypotheses are numerically verified in all the test cases, but they are not proved. This makes the asymptotic ratio property unproved, but numerically verified for all the tested cases, as shown below.

Tables 6.2, 6.3 and 6.3 give a sample of the ratio property obtained for some tested cases and gives also the ratio $R_L = \mathbf{1} + Z_L(s_L)$, corresponding to the limit problem of each case.

Obviously, the ratios $\{R_n\}$ become closer and closer of R_L, as n becomes larger.
Note that only the border and central components of the ratio vectors are given.
Besides, we also show, for different values of $\epsilon_{tol}^{n_s}$, the number of slices that should be solved sequentially for reaching the ratio property, up to $\epsilon_{tol}^{n_s}$.
In the experiments done for getting those tables, we used the following:
- space domain: $\Omega = [-1, 1] \subset \mathbb{R}$, - space discretization step: $h = 1/8$,
- initial condition: $u_0(x) = (1 - x^2)$, - initial time-step of integration: $\tau = \frac{h^2}{2}$,
- $a = 3$, - $S = 3$.

	Case A		Case B	
	p = 1	m = 0.9	p = 1	m = 0.5
n	$\mathbf{R_{n,border}}$	$\mathbf{R_{n,central}}$	$\mathbf{R_{n,border}}$	$\mathbf{R_{n,central}}$
1	2.83559541326472	3.99999999999997	1.44193431436541	3.99999999999998
2	3.99999999999998	3.99999999999998	3.99999992871993	3.99999999999995
3	3.99999999999999	3.99999999999999	3.99999999636228	3.99999999999994
4	3.99999999999993	3.99999999999993	3.99999999934348	3.99999999999993
5	3.99999999999999	3.99999999999999	3.99999999984815	3.99999999999996
6	3.99999999999990	3.99999999999990	3.99999999996271	3.99999999999993
7	3.99999999999998	3.99999999999998	3.99999999999065	3.99999999999991
8	3.99999999999996	3.99999999999996	3.99999999999762	3.99999999999994
9	3.99999999999989	3.99999999999989	3.99999999999940	3.99999999999998
10	3.99999999999996	3.99999999999996	3.99999999999982	3.99999999999997
⋮	⋮	⋮	⋮	⋮
99	3.99999999999990	3.99999999999990	3.99999999999999	3.99999999999999
100	3.99999999999998	3.99999999999998	3.99999999999999	3.99999999999999
	$\mathbf{R_{L,border} = 4}$	$\mathbf{R_{L,central} = 4}$	$\mathbf{R_{L,border} = 4}$	$\mathbf{R_{L,central} = 4}$

Table 6.2: Non Linear RD Problem - Asymptotic Ratio Property (1).

	Case C		Case D	
	p = 0.9 m = 0.6		p = 0.8 m = 0.6	
n	$\mathbf{R_{n,border}}$	$\mathbf{R_{n,central}}$	$\mathbf{R_{n,border}}$	$\mathbf{R_{n,central}}$
1	1.78364023119042	3.99999999999996	1.83352562285065	3.99999999999998
2	3.99999999999997	3.88317268196282	3.99999999999997	3.87222653761075
3	3.99999999999999	3.81487247996191	4.00000000000000	3.82121933675045
⋮	⋮	⋮	⋮	⋮
30	3.99999999999998	3.99072144146599	3.99999999999996	3.99804360152481
31	3.99999999999996	3.99263112282502	3.99999999999998	3.99869478299589
32	3.99999999999992	3.99414851379288	3.99999999999996	3.99913194536777
⋮	⋮	⋮	⋮	⋮
61	3.99999999999995	3.99999278730452	3.99999999999999	3.99999999667659
62	3.99999999999995	3.99999427527719	3.99999999999999	3.99999999786274
63	3.99999999999995	3.99999545628284	3.99999999999996	3.99999999862609
⋮	⋮	⋮	⋮	⋮
94	3.99999999999995	3.99999999647811	3.99999999999997	3.99999999999997
95	3.99999999999995	3.99999999720467	3.99999999999997	3.99999999999997
96	3.99999999999995	3.99999999778133	3.99999999999997	3.99999999999997
	$\mathbf{R_{L,border} = 4}$	$\mathbf{R_{L,central} = 4}$	$\mathbf{R_{L,border} = 4}$	$\mathbf{R_{L,central} = 4}$

Table 6.3: Non Linear RD Problem - Asymptotic Ratio Property (2).

	Case E		Case F	
	p = 0.7 m = 0.5		p = 0.6 m = 0.5	
n	$\mathbf{R_{n,border}}$	$\mathbf{R_{n,central}}$	$\mathbf{R_{n,border}}$	$\mathbf{R_{n,central}}$
1	1.61221552351503	3.99999999999999	1.63655354424360	3.99999999999999
2	3.99999999999999	3.74612487708889	3.99999999999999	3.86170558448441
3	3.99999999999999	3.64559494510411	3.99999999999999	3.83084363332960
⋮	⋮	⋮	⋮	⋮
20	3.99999999999996	3.99152300638580	3.99999999999999	3.82686086557656
21	3.99999999999999	3.99501827578468	3.99999999999998	3.85871127476992
22	3.99999999999997	3.99708879119099	3.99999999999998	3.88637404614876
⋮	⋮	⋮	⋮	⋮
61	3.99999999999999	3.99999999999874	3.99999999999997	3.99999715308105
62	3.99999999999998	3.99999999999926	3.99999999999997	3.99999784243600
63	3.99999999999997	3.99999999999955	3.99999999999999	3.99999836487060
⋮	⋮	⋮	⋮	⋮
86	3.99999999999999	3.99999999999999	3.99999999999999	3.99999999721978
87	3.99999999999999	3.99999999999999	3.99999999999999	3.99999999789297
	$\mathbf{R_{L,border} = 4}$	$\mathbf{R_{L,central} = 4}$	$\mathbf{R_{L,border} = 4}$	$\mathbf{R_{L,central} = 4}$

Table 6.4: Non Linear RD Problem - Asymptotic Ratio Property (3).

	$\epsilon_{tol}^{n_s}$	10^{-3}	10^{-6}	10^{-9}
Case A	n_s	3	3	3
Case B		3	3	5
Case C		34	64	94
Case D		30	47	63
Case E		24	36	49
Case F		36	61	86

Table 6.5: Non Linear RD Problem - Values of n_s for different $\epsilon_{tol}^{n_s}$.

6.3.2 Application of RaPTI

Tables (6.6) and (6.7) below summarize some results that were obtained by applying RaPTI algorithm on the reaction-diffusion problem, in the case of asymptotic similarity when $0 < m < p \leq 1$ and for 15 combinations of the problem parameters p and m.
When doing the preliminary analysis, the procedure FIND_RATIOS_MODEL, described in section 4.3.2 (of Chapter 4) and providing a mathematical model for predicting the ratios, could ultimately be repeated for every combination of the problem parameters (p and m), and for every change in the initial condition.
However, here also, the numerical experiments show that the rescaling technique and the resulting similarity and ratio properties make the same mathematical form of the model R_{fit}, with the same number n_{prec} of previous data to be used, hold the extrapolation test for all combinations of problem parameters and for any initial condition (within the same global behavior, of course), as well as at all iterations.
Hence, the procedure FIND_RATIOS_MODEL is implemented, once for all, in the preliminary stage, for an arbitrary combination (p, m) satisfying $0 < m < p \leq 1$ and for any initial condition yielding the same global behavior.
The resulting mathematical model that has been used for the predictions of the ratios is a polynomial of degree $d = 2$, obtained by fitting the last $n_{prec} = 3$ computed ratios, in the least-squares sense.

The total number N of slices, and therefore the interval of integration $[0, T]$, corresponds to the maximum number preventing the explosive solution from exceeding the machine capacity. The total number of iterations vary from one case to another, but in all cases, the results show how small is this number compared to the total number of slices. This ascertains the fast convergence of RaPTI when applied to the reaction-diffusion problem.

In the experiments done for getting tables (6.6) and (6.7), we used:
- space domain: $\Omega = [-1, 1] \subset \mathbb{R}$,
- space discretization step: $h = 1/8$,
- $a = 3$
- $S = 3$,
- initial condition: $u_0(x) = (1 - x^2)$,
- initial time-step of integration: $\tau = \frac{h^2}{2}$,
- $\epsilon_{tol}^{eos} = 10^{-14}$,
- $\epsilon_{tol}^{n_s} = \epsilon_{tol}^{g} = 10^{-8}$, $n_s^{min} = 8$.

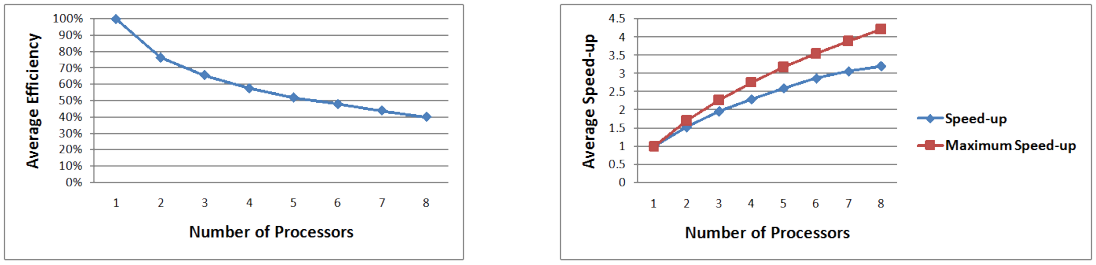
Case	1	2	3	4	5	6	7
p	**1**	**1**	**1**	**1**	**1**	**0.9**	**0.9**
m	**0.9**	**0.8**	**0.7**	**0.6**	**0.5**	**0.8**	**0.7**
T	243	239	239	238	238	1.9×10^{31}	2×10^{31}
N	**460**	**409**	**358**	**307**	**255**	**410**	**359**
$\mathbf{T_s}$	**19.53**	**19.54**	**16.91**	**16.42**	**14.57**	**19.82**	**19.33**
n_s	8	8	8	8	8	143	101
n_I	1	1	1	1	1	1	1
T_2	10.6	10.65	9.47	9.17	8	15.6	13.59
$\mathbf{E_2}$	**0.921**	**0.917**	**0.893**	**0.895**	**0.911**	**0.635**	**0.711**
$\mathbf{S_2}$	**1.842**	**1.835**	**1.786**	**1.791**	**1.821**	**1.271**	**1.422**
S_2^{max}	1.966	1.962	1.956	1.949	1.939	1.483	1.561
T_4	5.9	5.77	5.34	5.11	4.66	12.54	10.33
$\mathbf{E_4}$	**0.828**	**0.847**	**0.792**	**0.803**	**0.782**	**0.395**	**0.468**
$\mathbf{S_4}$	**3.310**	**3.386**	**3.167**	**3.213**	**3.127**	**1.581**	**1.871**
S_4^{max}	3.802	3.778	3.749	3.710	3.656	1.955	2.169
T_8	3.61	3.38	3.26	3.16	2.94	11.17	8.91
$\mathbf{E_8}$	**0.676**	**0.723**	**0.648**	**0.650**	**0.619**	**0.222**	**0.271**
$\mathbf{S_8}$	**5.410**	**5.781**	**5.187**	**5.196**	**4.956**	**1.774**	**2.169**
S_8^{max}	7.132	7.037	6.918	6.766	6.559	2.325	2.694

Table 6.6: Non Linear RD Problem - Application of RaPTI (1).

Case	8	9	10	11	12	13	14	15
p	**0.9**	**0.9**	**0.8**	**0.8**	**0.8**	**0.7**	**0.7**	**0.6**
m	**0.6**	**0.5**	**0.7**	**0.6**	**0.5**	**0.6**	**0.5**	**0.5**
T	2.03×10^{31}	2.1×10^{31}	5.9×10^{61}	6.2×10^{61}	6.7×10^{61}	2.6×10^{92}	2.9×10^{92}	1.4×10^{123}
N	**308**	**257**	**359**	**308**	**257**	**308**	**257**	**257**
n_s	84	70	112	58	43	94	45	78
n_I	1	1	3	1	1	9	1	4
$\mathbf{T_s}$	**17.76**	**16.76**	**21.17**	**19.7**	**19.58**	**22.16**	**21.85**	**26.21**
T_2	13.01	11.83	16.19	13.58	12.45	17.68	14.52	21.13
$\mathbf{E_2}$	**0.683**	**0.708**	**0.654**	**0.725**	**0.786**	**0.627**	**0.752**	**0.620**
$\mathbf{S_2}$	**1.365**	**1.417**	**1.308**	**1.451**	**1.573**	**1.253**	**1.505**	**1.240**
S_2^{max}	1.571	1.572	1.524	1.683	1.713	1.532	1.702	1.534
T_4	9.92	9.21	12.78	9.53	8.59	14.47	10.25	17.15
$\mathbf{E_4}$	**0.448**	**0.455**	**0.414**	**0.517**	**0.570**	**0.383**	**0.533**	**0.382**
$\mathbf{S_4}$	**1.790**	**1.820**	**1.656**	**2.067**	**2.279**	**1.531**	**2.132**	**1.528**
S_4^{max}	2.200	2.201	2.066	2.556	2.663	2.088	2.622	2.094
T_8	8.37	7.91	11.31	7.64	6.85	12.97	8.35	15.26
$\mathbf{E_8}$	**0.265**	**0.265**	**0.234**	**0.322**	**0.357**	**0.214**	**0.327**	**0.215**
$\mathbf{S_8}$	**2.122**	**2.119**	**1.872**	**2.579**	**2.858**	**1.709**	**2.617**	**1.718**
S_8^{max}	2.750	2.752	2.513	3.451	3.685	2.551	3.594	2.560

Table 6.7: Non Linear RD Problem - Application of RaPTI (2).

Actually, RaPTI has been tested on the previous 15 cases, using 2, 3, 4, 5, 6, 7, and 8 processors.
Figure 6.3.2 shows how the averaged values of the efficiency and speed-up vary with the number of processors.



(a) Efficiency (b) Speed-up

Figure 6.5: Non Linear RD Problem - Efficiencies and speed-ups.

Besides, figure (6.6) illustrates the speed-up, while using 8 processors, and the corresponding upper bound in each of the 15 cases. It shows clearly the high speed-ups in the cases 1 to 5 (corresponding to $p = 1$), due to a quick stabilization of the ratios.

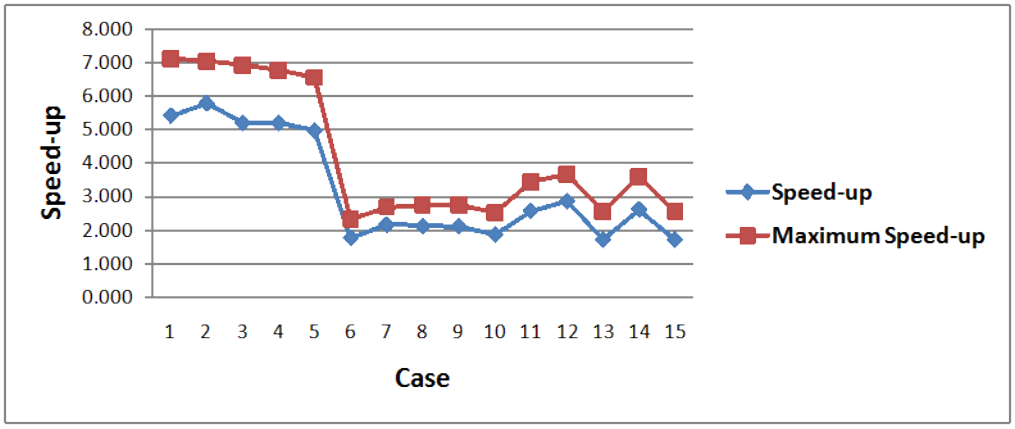


Figure 6.6: Non Linear RD Problem - Speed-up Using 8 Cores

6.4 Numerical Results: Linear case

The particular case $m = p = 1$, simplifies the reaction-diffusion problem to a linear one, similar to that discussed in chapter 2, with the same explosive global behavior as that described for the considered general case $0 < m \leq p \leq 1$.
The differential equation reduces to:

$$\frac{\partial u}{\partial t} = \Delta u + au.$$

A space semi-discretization of dimension K, yields then the equivalent initial value problem:

$$\begin{cases} \frac{dY}{dt} = -AY + aY, \, t > 0 \\ Y(0) = Y_0, \end{cases}$$

where $A \in \mathbb{R}^{K\times K}$ is a sparse symmetric positive definite matrix that discretizes the operator $-\Delta$.
The problem has now the general form of the linear first order initial value problem (3.10.1), given in Chapter 3, in which one seeks $Y : [0, T] \to \mathbb{R}^k$, such that:

$$(S) \qquad \begin{cases} \frac{dY}{dt} = BY, \qquad t > 0, & (6.34.1) \\ Y(0) = Y_0 > 0. & (6.34.2) \end{cases} \tag{6.34}$$

where:$B = (aI - A)$ is a constant matrix in $\mathbb{R}^{K\times K}$, I being the identity matrix.

Using the EOS condition (6.10) and rescaling with the critical choice (1.40) for β_n make solving problem (6.34) equivalent to solving, on each n^{th} slice $[0, s_n]$, corresponding to $[T_{n-1}, T_n]$, an initial value shooting problem:

$$\begin{cases} \frac{dZ_n}{ds} = A_n\,(\mathbf{1} + Z_n) \;=\; G_n(Z_n), \;\; 0 < s < s_n & (6.35.1) \\ Z_n(0) = 0, & (6.35.2) \\ \forall s < s_n,\; \|Z_n(s)\|_\infty < S, \text{ and } \|Z_n(s_n)\|_\infty = S. & (6.35.3) \end{cases} \tag{6.35}$$

where $\mathbf{1}$ is the vector of ones in $\mathbb{R}^K$, and:

$$A_n = D_n^{-1} A D_n \tag{6.36}$$

is a matrix in $\mathbb{R}^{K\times K}$, constant on each n^{th} slice, since D_n depends only on the starting value Y_{n-1} at the n^{th} slice.

Proposition 9 ***: Asymptotic Similarity***
If the eigenvalue λ of A having the largest real part is unique, with u an eigenvector corresponding to λ, and if the initial condition satisfies $u^* A Y_0 \neq 0$, then the rescaled systems (3.18) are asymptotically similar to the limit problem:

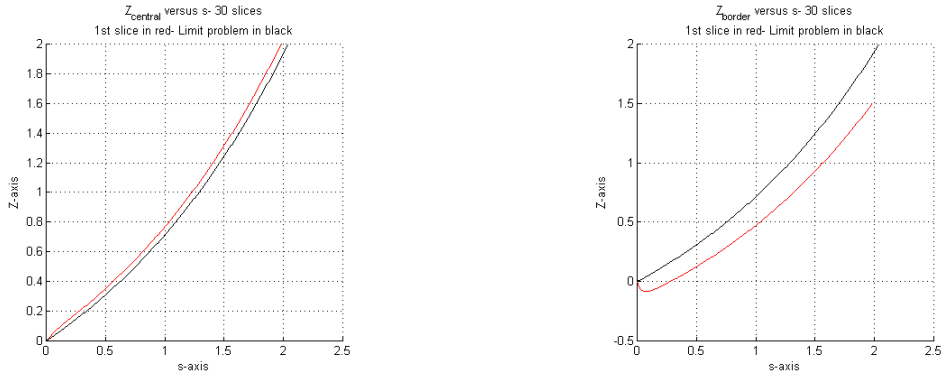
$$\begin{cases} \frac{dZ_L}{ds} = G_L(Z_L), \;\; 0 < s \leq s_L, & \textit{(6.37.1)} \\ Z_L(0) = 0, & \textit{(6.37.2)} \\ \forall s < s_n,\; \|Z_n(s)\|_\infty < S, \;\textit{and}\; \|Z_n(s_n)\|_\infty = S. & \textit{(6.37.2)} \end{cases} \tag{6.37}$$

where:

$$G_L(Z_L) = \lambda \exp(s\lambda)\,\mathbf{1},$$

Proof:
The proof follows directly from proposition 6, given in Chapter 3, together with the nonzeroness condition, that is known to be satisfied in this reaction-diffusion problem, and the choice $\beta_n = 1, \;(\forall n)$. ∎

Figure 6.7 illustrates this asymptotic similarity: the rescaled solution $Z_n(s)$ on successive time-slices are represented on the same graph (central and border component only). Each curve, corresponding to the n^{th} slice $[0, s_n]$, starts at $Z_{n,i}(0) = 0$ and ends at $Z_{n,i}(s_n)$.
The curve corresponding to the first slice is in red and that corresponding to the limit problem is in black. All others are in green.
One can notice how quick (starting from the second slice) the curves superpose on that corresponding to the limit problem (in black).
Zooming out does not even allow, in this case, to point out the intermediate curves in green!



(a) Central Component (b) Border Component

Figure 6.7: Linear RD Problem - Z versus s, on 30 slices, $m = p = 1$, $a = 3$, $S = 2$, with $\Omega = [-1, 1] \subset \mathbb{R}$ and $u_0(x) = 1 - x^2$.

6.4.1 Existence of a Ratio Property

Tables 6.8 and 6.9 give a sample of the asymptotic ratio property obtained for some tested cases and gives also the ratio $R_L = \mathbf{1} + Z_L(s_L)$, corresponding to the limit problem. Note that only two components of the ratio vectors are given.

In the experiments done for getting those tables, we used the following:
- space domain: $\Omega = [-1, 1] \subset \mathbb{R}$,
- space discretization step: $h = 1/8$,
- $S = 2$,
- $\epsilon_{tol}^{eos} = 10^{-15}$,
- initial time-step of integration: $\tau = \frac{h^2}{2}$.

The initial condition $u_0(x)$ is respectively:
(i) symmetric, in table 6.8, and equal to:

$$u_1(x) = (1 - x^2) \tag{6.38}$$

(ii) non symmetric, in table 6.8, and equal to:

$$u_2(x) = \begin{cases} 2x + 2, & \text{if } -1 \leq x \leq -\frac{1}{2} \\ -\frac{2}{3}x + \frac{2}{3}, & \text{if } -\frac{1}{2} \leq x \leq 1 \end{cases} \tag{6.39}$$

	Case A $a = 3$ $u_0(x) = u_1(x)$		Case B $a = 3.3$ $u_0(x) = u_1(x)$	
n	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,central}}$	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,central}}$
1	2.49715612019833	2.99999999999999	2.49715603875983	2.99999999999999
2	2.99999999999999	2.99999999806807	2.99999999999824	2.99999990022867
3	2.99999999999999	2.99999999999999	2.99999999999985	2.99999999999999
4	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
5	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
6	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
⋮	⋮	⋮	⋮	⋮
29	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
30	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
	$\mathbf{R_{L,1} = 3}$	$\mathbf{R_{L,central} = 3}$	$\mathbf{R_{L,1} = 3}$	$\mathbf{R_{L,central} = 3}$

Table 6.8: Linear RD Problem - Asymptotic Ratio Property (1).

	Case C $a = 3$ $u_0(x) = u_2(x)$		Case D $a = 3.3$ $u_0(x) = u_2(x)$	
n	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,central}}$	$\mathbf{R_{n,1}}$	$\mathbf{R_{n,central}}$
1	1.00113381871731	1.92337614706756	1.01378183327573	1.93541743577233
2	2.99660239696541	2.99814457356693	2.95921903654898	2.97949271999232
3	2.99999999914856	2.99999999956283	2.9999975002082	2.99999875013544
4	2.99999999999999	2.99999999999999	2.9999999998568	2.9999999999284
5	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
6	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
⋮	⋮	⋮	⋮	⋮
29	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
30	2.99999999999999	2.99999999999999	2.99999999999999	2.99999999999999
	$\mathbf{R_{L,1} = 3}$	$\mathbf{R_{L,central} = 3}$	$\mathbf{R_{L,1} = 3}$	$\mathbf{R_{L,central} = 3}$

Table 6.9: Linear RD Problem - Asymptotic Ratio Property (2).

6.4.2 Application of RaPTI

Table (6.10) below summarizes some results that are obtained by applying RaPTI algorithm on the linear reaction-diffusion problem (when $m = p = 1$), for 4 different values of the problem parameter a and for two different initial conditions: $u_0(x) = u_1(x)$ and $u_0(x) = u_2(x)$, where u_1 and u_2 are given in (6.38) and (6.39).
The mathematical model that has been used for the predictions of the ratios is a polynomial of degree $d = 2$, obtained by fitting the last $n_{prec} = 3$ computed ratios, in the least-squares sense.

The total number N of slices, and therefore the interval of integration $[0, T]$, corresponds to the maximum number preventing the explosive solution from exceeding the machine capacity.

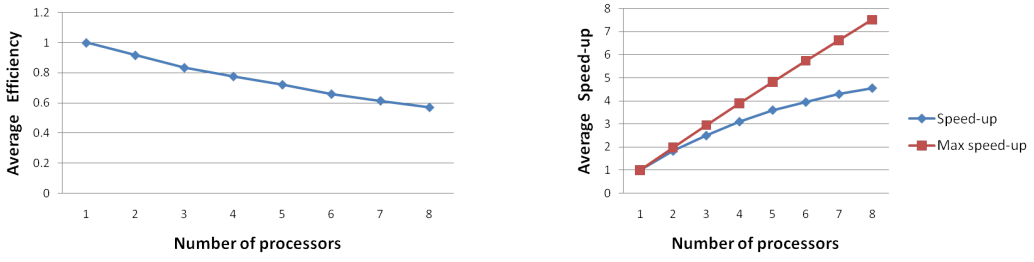
The results show that, when applied to those 8 linear cases, RaPTI algorithm converges in only one iteration!

In the experiments done for getting table (6.10), we used the following:
- space domain: $\Omega = [-1, 1] \subset \mathbb{R}$,
- space discretization step: $h = 1/8$,
- initial time-step of integration: $\tau = \frac{h^2}{2}$,
- $S = 2$,
- $\epsilon_{tol}^{eos} = 10^{-15}$,
- $\epsilon_{tol}^{g} = 10^{-8}$,
- $\epsilon_{tol}^{n_s} = 10^{-8}$,
- $n_s^{min} = 6$.

Case	1	2	3	4	5	6	7	8
	$\mathbf{u_0(x) = u_1(x)}$				$\mathbf{u_0(x) = u_2(x)}$			
a	**3**	**3.1**	**3.2**	**3.3**	**3**	**3.1**	**3.2**	**3.3**
T	1315.86	1110.74	961.21	847	1316.41	1109.87	960.36	846.43
N	**646**	**646**	**646**	**646**	**646**	**646**	**646**	**646**
$\mathbf{T_s}$	**5.96**	**5.4**	**4.61**	**4.31**	**5.94**	**5.33**	**4.6**	**4.3**
n_s	6	6	6	6	6	6	6	6
n_I	1	1	1	1	1	1	1	1
T_2	3.21	2.92	2.52	2.40	3.17	2.88	2.53	2.40
$\mathbf{E_2}$	**0.93**	**0.92**	**0.91**	**0.90**	**0.94**	**0.93**	**0.91**	**0.90**
$\mathbf{S_2}$	**1.86**	**1.85**	**1.83**	**1.80**	**1.87**	**1.85**	**1.82**	**1.79**
S_2^{max}	1.98	1.98	1.98	1.98	1.98	1.98	1.98	1.98
T_4	1.86	1.70	1.52	1.43	1.86	1.67	1.53	1.43
$\mathbf{E_4}$	**0.80**	**0.79**	**0.76**	**0.75**	**0.80**	**0.80**	**0.75**	**0.75**
$\mathbf{S_4}$	**3.20**	**3.18**	**3.03**	**3.01**	**3.19**	**3.19**	**3.01**	**3.01**
S_4^{max}	3.89	3.89	3.89	3.89	3.89	3.89	3.89	3.89
T_8	1.23	1.12	1.05	1.01	1.23	1.13	1.06	1.02
$\mathbf{E_8}$	**0.61**	**0.60**	**0.55**	**0.53**	**0.60**	**0.59**	**0.54**	**0.53**
$\mathbf{S_8}$	**4.85**	**4.82**	**4.39**	**4.27**	**4.83**	**4.72**	**4.34**	**4.22**
S_8^{max}	7.51	7.51	7.51	7.51	7.51	7.51	7.51	7.51

Table 6.10: Linear RD Problem - Application of RaPTI.

RaPTI has been tested on the previous 8 cases, using 2, 3, 4, 5, 6, 7, and 8 processors. Figure 6.4.2 shows how the values of the efficiency and speed-up, averaged on the 8 cases, vary with the number of processors.



(a) Efficiency (b) Speed-up

Figure 6.8: Linear RD Problem - Efficiencies and speed-ups.

Remark 10
One can notice that the resulting speed-ups, in those linear cases, are not close enough to the maximum speed-ups (stated by Amdahl's law), in spite of:

- *the very strong ratio property (yielding $n_s = n_s^{min} = 6$) and*
- *the perfect convergence of the algorithm, in only one iteration.*

Those two properties are, usually, ideal conditions for getting very high speed-ups that are much close to the maximum ones.
We suspect the very small times of execution (of order 5 seconds, in the sequential computations) to block the speeding-up of the parallel computations.

Chapter 7

Satellite Trajectories

Spatial vehicles need locating systems. The adapted orbit model is a system of second order differential equations and needs all through the mission to regularly be updated (see [7], [62]). This "orbit restoration" aims at taking into account any modification of the trajectory as well as the imperfections of the orbit model. For reaching a high accuracy, a mass of computations that are extremely time-consuming are required, due to the big amount of perturbations that must be taken into account.
This makes time parallelism well suited for tackling such real-time problems. Erhel and Rault proposed, in [7] and [62], an interesting parallel method for solving the problem, namely a multiple shooting method, and all perturbing forces were taken into consideration.

Since the computation of the different perturbing forces requires some specialized programs, we propose to apply RaPTI Algorithm, in this Chapter, to a simplified model of the problem.

In section 1, we start by deriving the simplified J_2-model that takes into consideration the central earth attraction in addition to a unique perturbation: the one deriving from the flattening of the earth. We give the explicit differential equations modeling the resulting motion and describe the effects of the J_2-perturbation.
In section 2, we define the rectangular coordinate system that we use and translate the problem, in this coordinate system, to a first order system of ODE's.
In section 3, we apply the sequential RaTI algorithm onto the case of invariance that corresponds to a Keplerian motion (with no perturbation).
In section 4, we show how our method yields a weak ratio property, in the J_2-perturbed motion, and we apply RaPTI Algorithm. It yields significant speed-ups.

The notations used in this chapter are those indicated in table 10.1.

7.1 Derivation of a Simplified Satellite Model

7.1.1 Differential Equations Describing the Problem

Below are described some basic results concerning satellite trajectories. Our main reference is that of O.Zarrouati [63].

General Equation of a Satellite Motion:
Satellite trajectories are computed by solving a system of second order differential equations that follows from the general equation of motion given by Newton's second law:

$$\vec{F} = m\vec{\ddot{r}} \tag{7.1}$$

where:
m is the mass of the satellite,
$\vec{r}$ is the position-vector of the satellite (radius vector from earth's center)
$\vec{\ddot{r}}$ is the acceleration vector of the satellite, and
$\vec{F}$ is the resultant vector-force applied on the satellite.

Keplerian Motion:
In the proximity of the earth, it can be considered that the gravitational attraction, assumed to be centered, is the only force that is applied to the satellite:

$$\vec{F} = -GM\frac{m}{\|\vec{r}\|^3}\vec{r} = -\mu\frac{m}{\|\vec{r}\|^3}\vec{r} \tag{7.2}$$

where G is the universal gravitational constant, M the earth's mass and

$$\mu = GM = 3986005 \times 10^8 m^3/s^2. \tag{7.3}$$

Then, equation (7.1) reduces to the following system of differential equations which is characteristic of a Keplerian motion:

$$\vec{\ddot{r}} = -\frac{\mu}{\|\vec{r}\|^3}\vec{r}. \tag{7.4}$$

In such a case, the satellite moves in a fixed plane through the center of the earth, along an ellipse having this center as one focus. The "Keplerian plane" is completely defined by the initial position and velocity vectors of the satellite.
If a and b denote the semi-major and semi-minor axes respectively, then the eccentricity of the elliptical orbit is $e = \frac{\sqrt{a^2-b^2}}{a}$, and its semi-latus rectum is $p = \frac{b^2}{a} = a(1-e^2)$.

- Kepler's first law gives the polar equation of the orbital ellipse:

  $$r = \frac{p}{1 + e\cos\theta},$$

  where r is the distance to the center of the earth and θ is the polar angle determined when the origin is at the earth's center and $\theta = 0$ along the major axis directed toward the perigee,

- Kepler's second law gives: $r^2\dot{\theta} = \sqrt{\mu a(1-e^2)}$,

- Kepler's third law gives: $n^2a^3 = \mu$, where $n = \frac{2\pi}{P}$ is the mean motion of the satellite (i.e. mean angular velocity) and P is the period of the the satellite.

Thus, a Kelplerian motion is periodic of period:

$$P = 2\pi\sqrt{\frac{a^3}{\mu}}. \tag{7.5}$$

Perturbing Forces:
In fact, many other forces perturb the satellite motion (the least that can be cited is that the gravitational attraction is not centered!).
Some of these perturbation forces derive from gravitational potentials (of the earth, the moon, the sun,...) and some others are surface forces (atmospheric frictional force, solar radiation pressure,...).
All these forces are detailed by Rault in [62] and their order of magnitude, compared to that of the centered attraction of the earth, is estimated as follows:

Centered gravitational attraction of the earth	:	1
Force due to the flattening of the earth	:	10^{-3}
Force due to other geometric irregularities of the earth	:	10^{-6}
Force due to atmospheric frictions	:	10^{-9} to 10^{-5}
Force due to the moon gravitational attraction	:	10^{-7}
Force due to the sun gravitational attraction	:	10^{-8}
Force due to the solar radiation pressure	:	10^{-9}

In [62], Rault considered all these forces and relied, in their calculation, on programs provided by the CNES (Centre National des Etudes Spatiales).

In this thesis, we will only consider the perturbation due to the flattening of the earth since it causes (by far) the most important perturbation of the motion. Moreover, numerical approximations of this perturbation can be given explicitly.
However, the numerical method we are devising, can be extended to other perturbing forces if one has available programs for their calculation.

General Earth's Gravitational Potential:
The earth's flattening comes from the zonal harmonic J_2 which causes the ellipticity of the earth (whose linear effect is $a - b \approx 20$ km) making the earth's shape like an ellipsoid rather than a sphere.
But the earth's shape is not exactly an ellipsoid: it is rather shaped like a pear. This pear shape is caused by the spherical harmonic J_3. However, its effect at the North and South Poles, is of order 30 m only.
The gravitational potential taking into consideration all the geometric irregularities of the earth can be expanded into a series of spherical harmonics:

$$\begin{aligned} U = \frac{\mu}{r}\Big\{1 + \sum_{n=1}^{\infty}\left(\frac{r_{eq}}{r}\right)^n J_n P_n(\sin\varphi) + \\ \sum_{n=1}^{\infty}\sum_{m=1}^{n}\left(\frac{r_{eq}}{r}\right)^n \left[C_{nm}\cos(m\lambda) + S_{nm}\sin(m\lambda)\right] P_{nm}(\sin\varphi)\Big\} \end{aligned} \tag{7.6}$$

where r is the magnitude of the position vector $\vec{r}$, r_{eq} the equatorial radius,
φ the geocentric latitude, λ the longitude,
P_k the Legendre polynomials of degree k,
$P_{n,m}$ the Legendre functions of degree n and order m,
J_n the zonal harmonics of order n (with $J_1 = 0$),
S_{nm} and C_{nm} the tesseral harmonics of degree n and order m.

The effects of the zonal harmonics on satellite orbits is much greater than that of the tesseral harmonics. Only zonal harmonics (J_2, J_3, J_4,...) will give observable variations of the orbital elements themselves.
The tesseral harmonics cause oscillatory disturbances that rapidly change their sign, whereas the effect of zonal harmonics is cumulative.
The *earth's flattening*, due to the *zonal harmonic* J_2, causes the largest deviation of the earth's gravitational field from that of a homogeneous sphere.

The J_2-Perturbed Earth's Gravitational Potential:
Hence, the J_2-model takes into consideration the central earth attraction in addition to a unique perturbation: the one deriving from the flattening of the earth (which is, in fact, the dominant one), neglecting all other perturbations.
In this case, the earth's gravitational potential function is given by:

$$u(J_2) = \frac{\mu}{r}\left\{1 + \left(\frac{r_{eq}}{r}\right)^2 J_2 P_2(\sin\varphi)\right\}, \tag{7.7}$$

where:
$\mu = 3986005 \times 10^8 m^3/s^2$, as given in (7.3),
the equatorial radius r_{eq} is:

$$r_{eq} = 6378.137km, \tag{7.8}$$

the zonal harmonic J_2, of order 2, is equal to:

$$J_2 = -11.10^{-4}, \tag{7.9}$$

$r = \|\vec{r}\|_2$ is the magnitude of the position vector,
$P_2(\sin\varphi)$ is the second degree Legendre polynomial, in the variable $\sin\varphi$ (φ being the latitude):

$$P_2(\sin\varphi) = \frac{3}{2}\sin^2\varphi - \frac{1}{2}, \tag{7.10}$$

Differential Equation Modelizing the J_2-Problem:
Since $\vec{\nabla}u(J_2)$ is the force deriving from this gravitational potential, per mass unit, then the force applied to the satellite is: $\vec{F} = m\vec{\nabla}u(J_2)$, and the general equation of the motion (7.1) leads to a second-order differential equation:

$$\ddot{\vec{r}} = \vec{\nabla}u(J_2) \tag{7.11}$$

It is then necessary to specify a set of initial conditions to completely determine the trajectory of the orbit.

7.1.2 Reference Coordinate Systems

Below, are presented two classical coordinate systems that will be used, further along in this chapter.

- **Earth Centered Inertial Coordinate Frame (ECI):**
  Also called "Geocentric Equatorial Inertial" system (GEI), it is an earth-based cartesian coordinate system $\left(O, \vec{i}, \vec{j}, \vec{k}\right)$, earth centered, where:
  - The X-axis points from the center of the Earth through the equator at a fixed spot in space called "the first point in Aries", or the "the vernal equinox" (i.e.

the position of the sun at the vernal equinox). This direction is the intersection of the Earth's equatorial plane and the ecliptic plane,

- The Z-axis is parallel to the rotation axis of the Earth, and
- The Y-axis completes the right-handed orthogonal set, i.e. $\vec{j} = \vec{k} \times \vec{i}$.

These axes are fixed in the center of the Earth and their directions are fixed in space. This means that every point on the Earth (except for the poles) rotate 360° around the z-axis every day! Furthermore, although the directions of the axes are fixed, the center of the coordinate system is not, as it moves around the sun every 365 days. Thus the ECI coordinate system is not really inertial ("fixed" in space) but is a moving coordinate system. For this reason it is called an Earth Centered Inertial coordinate frame. It is sort of fixed, so long as we stay very close to the Earth. When we are dealing with Earth orbiting satellites, virtually no accuracy at all is lost by treating this coordinate system as truly fixed in space; however, when studying interplanetary trajectories, we must account for the Earth's motion.

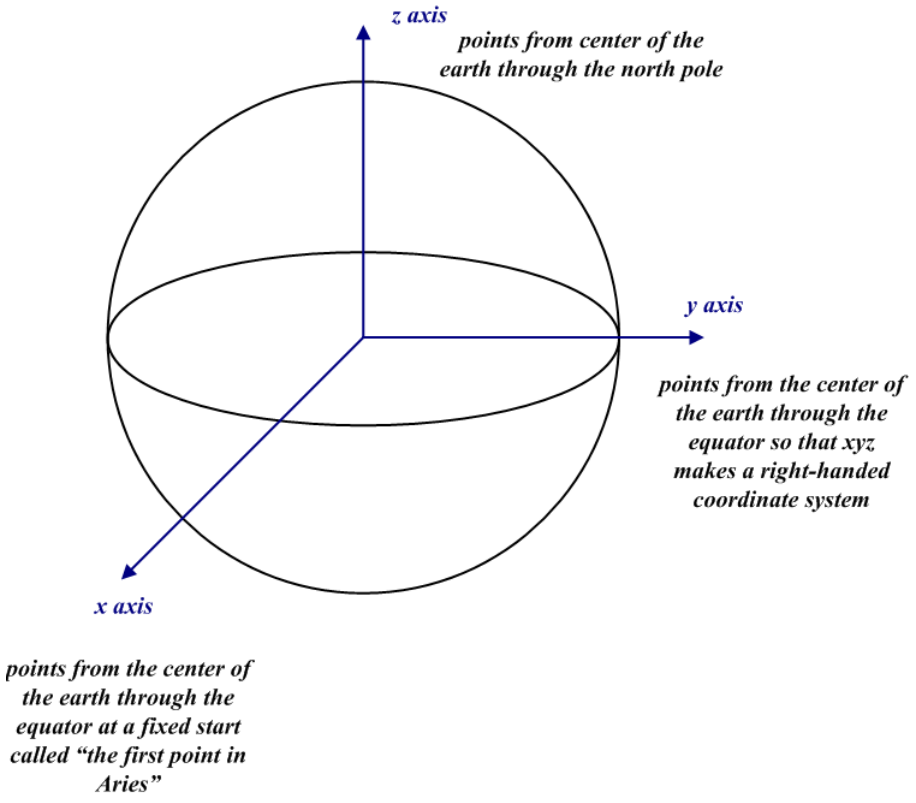


Figure 7.1: Earth Centered Inertial Coordinate Frame (ECI)

- **Perifocal Coordinate Frame (PQW):**
  This is a satellite-based cartesian coordinate system $\left(O, \vec{P}, \vec{Q}, \vec{W}\right)$ for an earth's satellite orbit, earth centered, whose rectangular axes are defined as follows:
  - the X-axis $(O, \vec{P})$ points from the center of the Earth toward the direction of perigee in the elliptical orbital plane,
  - the Y-axis $(O, \vec{Q})$ lies in the plane of the orbit and points from the center of the earth in a direction 90° advanced from the direction of perigee,

– the Z-axis $(O, \vec{W})$ points from the center of the earth in a direction perpendicular to the orbital plane in such a way that the triad $\vec{P}, \vec{Q}, \vec{W}$ forms a right-handed coordinate system, i.e. $\vec{W} = \vec{P} \times \vec{Q}$.

It is to be noticed that, in a perturbed (non-Keplerian) motion, we deal with a moving orbital plane, defined instantaneously by $\left(\vec{r}(t), \vec{\dot{r}}(t)\right)$*, so that the PQW-Frame is a moving frame in which* $\vec{P}$ *and* $\vec{Q}$ *lie in the instantaneous orbital plane, while* $\vec{W}$ *is normal to it.*

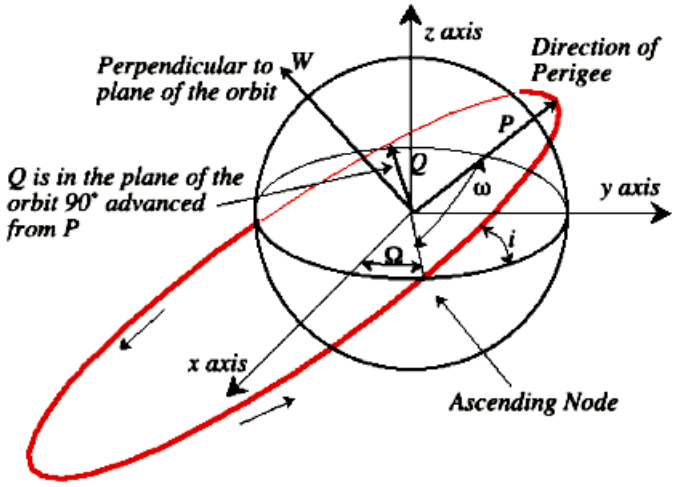


Figure 7.2: Perifocal Coordinate Frame (PQW)

7.1.3 Variables of the Motion

The motion of the satellite can be completely described by 6 independent variables.
In general, they consist of classical Cartesian variables or of Keplerian orbital elements. Actually, any analytic analysis of the problem uses the orbital elements, whereas numerical computation uses Cartesian variables.
In our computations, we will consider the orbital elements for defining the initial conditions, only.

- **Cartesian variables:**
  An obvious set of parameters is $(x, y, z, \dot{x}, \dot{y}, \dot{z})$, made of the cartesian components (x, y, z) of the position vector $\vec{r}$ and the cartesian components $(\dot{x}, \dot{y}, \dot{z})$ of the velocity vector $\vec{\dot{r}}$.

- **Orbital Elements**: Another possible set of parameters is the set of six Keplerian orbital elements $(a, e, i, \omega, \Omega, M)$, with:

  – a: semi-major axis of the orbit,

  – e: eccentricity of the orbit,

  – i: inclination of the orbit, i.e. the angle between the earth's equatorial plane and the orbital plane,

  – Ω: longitude (or right ascendant) of the ascending node, i.e. the angle measured in the equatorial plane between the x-axis and the intersection of the satellite's

orbit and equatorial plane as it moves from the southern hemisphere to the northern hemisphere (this point of intersection is called the ascending node of the orbit),

- ω: argument of perigee, i.e. the angle measured in the plane of the orbit between the ascending node and the perigee, and
- M: the mean anomaly, i.e. the fraction of the orbital period that has elapsed since the last passage at perigee, expressed as an angle. Alternatively, the eccentric anomaly E and the true anomaly θ can be used and the three anomalies are related by $M = E - e.\sin E$ and $\cos\theta = \frac{\cos E - e}{1 - e.\cos E}$.

Remark 11 *It is also convenient to define the mean motion n that depends directly on a:*

$$n = \sqrt{\frac{\mu}{a^3}}$$

and represents the mean angular velocity.
Note that, in a Keplerian motion, all the orbital elements are constant except for the mean anomaly that varies linearly with respect to time:

$$M = M_0 + n(t - T_0).$$

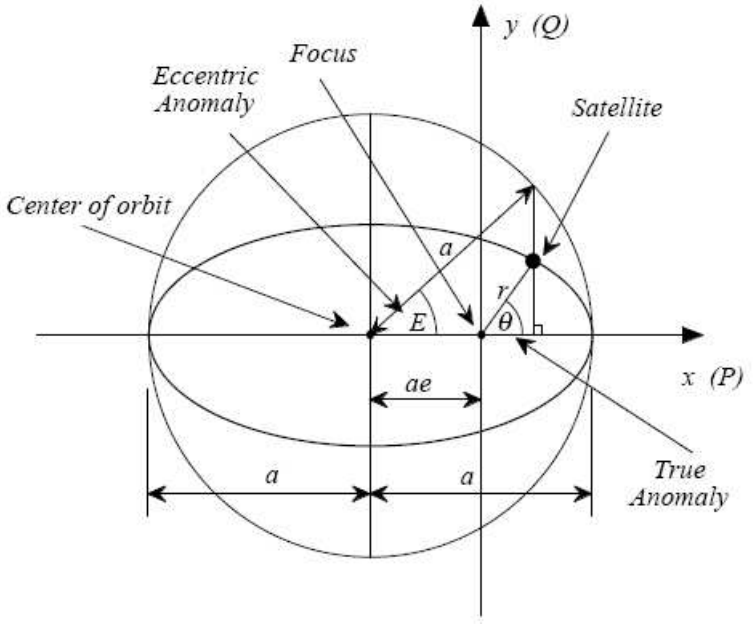


Figure 7.3: Satellite Orbital Elements (a, e, E, θ)

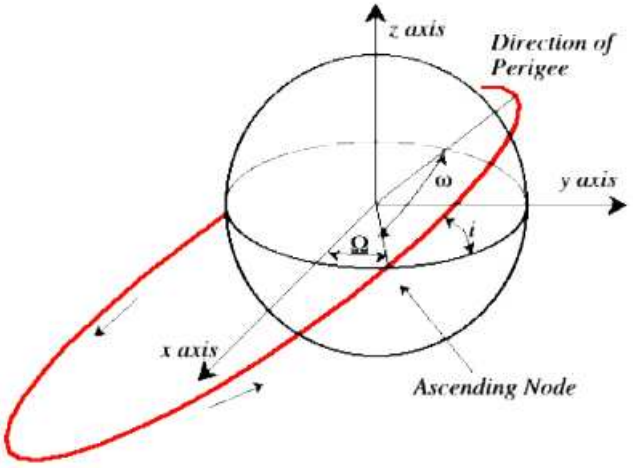


Figure 7.4: Satellite Orbital Elements (i, ω, Ω)

Osculating Ellipse:
Even if the satellite is subject to perturbing forces, we may still consider the satellite orbit as an ellipse but then, the parameters of this ellipse (the orbital elements) will no longer be constant: at each instant, this ellipse will be slightly different.
Imagine that at the instant under consideration all perturbing forces suddenly vanish, then the satellite will continue its motion along an exact ellipse: this is the *osculating ellipse*.

7.1.4 Coordinate Transformations

Here also, are described some basic results given in Zarrouati [63].
Those coordinate transformations will be used, in this chapter, for translating any initial condition, given by its six orbital elements, into its Cartesian coordinates and for converting coordinates that are explicitly given in the ECI-frame, into coordinates in a PQW-frame.

Orbital Elements $\longrightarrow$ PQW-coordinates:
Since $\vec{P}$ and $\vec{Q}$ lie in the orbital plane, it is easily shown that, given a set of 6 orbital elements $(a, e, i, \Omega, \omega, M)$, the corresponding position and velocity vectors in the PQW-frame are:

$$\vec{r}_{PQW} = \begin{pmatrix} a(\cos E - e) \\ a\sqrt{1-e^2}\sin E \\ 0 \end{pmatrix} \quad \text{and} \quad \dot{\vec{r}}_{PQW} = \begin{pmatrix} -\frac{na\sin E}{1-e\cos E} \\ \frac{na\sqrt{1-e^2}\cos E}{1-e\cos E} \\ 0 \end{pmatrix} \tag{7.12}$$

where $n = \sqrt{\frac{\mu}{a^3}}$ is the mean motion of the satellite.

PQW-coordinates $\longleftrightarrow$ ECI-coordinates:
Since the orbital elements i, Ω and ω are all measured from the axes of the inertial coordinate frame ECI, the easiest way to derive the coordinate transformation from the ECI-frame to the PQW-frame is via rotation matrices, by performing (in an equivalent way) the following three rotations, in succession:

- $r_Z(\Omega)$: rotate by Ω about the Z-axis,

- $r_X(i)$: rotate by i about the new X-axis,
- $r_Z(\omega)$: rotate by ω about the newer Z-axis.

Let $R_Z(\Omega)$, $R_X(i)$ and $R_Z(\omega)$ be the rotation matrices of these 3 rotations, respectively. The base vectors $\left(\vec{P}, \vec{Q}, \vec{W}\right)$ of the PQW-frame are the images of the vectors $\left(\vec{i}, \vec{j}, \vec{k}\right)$ by the successive 3 rotations and satisfy:

$$\begin{pmatrix} \vec{P} \\ \vec{Q} \\ \vec{W} \end{pmatrix} = A \begin{pmatrix} \vec{i} \\ \vec{j} \\ \vec{k} \end{pmatrix} \tag{7.13}$$

where A is the following transformation matrix, given in terms of the orbital elements i, Ω and ω by:

$$A = \begin{pmatrix} \cos\omega.\cos\Omega - \sin\omega.\cos i.\sin\Omega & \cos\omega.\sin\Omega + \sin\omega.\cos i.\cos\Omega & \sin\omega.\sin i \\ -\sin\omega.\cos\Omega - \cos\omega.\cos i.\sin\Omega & -\sin\omega.\sin\Omega + \cos\omega.\cos i.\cos\Omega & \sin i.\cos\omega \\ \sin i.\sin\Omega & -\sin i.\cos\Omega & \cos i \end{pmatrix}. \tag{7.14}$$

Therefore, and knowing that the transformation matrix A is orthogonal ($A^{-1} = A^T$), the cartesian coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2) of any point (or vector) of the space, in the ECI and PQW frames respectively, are then related by:

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}_{PQW} = A \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}_{ECI} \tag{7.15}$$

or equivalently:

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}_{ECI} = A^T \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}_{PQW} \tag{7.16}$$

Orbital Elements $\longrightarrow$ ECI-coordinates:
Combining relations (7.12) and (7.13) leads to the transformation from orbital elements to coordinates in the ECI-frame of the position and velocity vectors:

$$\begin{cases} \vec{r}_{ECI} = a(\cos E - e)\ \vec{P}_{ECI} + a\sqrt{1-e^2}\sin E\ \vec{Q}_{ECI} \\ \dot{\vec{r}}_{ECI} = -\frac{na\sin E}{1-e\cos E}\ \vec{P}_{ECI} + \frac{na\sqrt{1-e^2}\cos E}{1-e\cos E}\ \vec{Q}_{ECI} \end{cases} \tag{7.17}$$

where $\vec{P}_{ECI}$, $\vec{Q}_{ECI}$ and $\vec{W}_{ECI}$ are found, using (7.13).

ECI-coordinates $\longrightarrow$ Orbital Elements
Given cartesian coordinates in the ECI frame of the position and velocity vectors: $\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ r_3 \end{pmatrix}$ and $\dot{\vec{r}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = \begin{pmatrix} \dot{r}_1 \\ \dot{r}_2 \\ \dot{r}_3 \end{pmatrix}$

1. First compute: $r^2 = x^2 + y^2 + z^2$ and $v^2 = \dot{x}^2 + \dot{y}^2 + \dot{z}^2$
2. Deduce the semi-major axis a using:

$$\frac{1}{a} = \frac{2}{r} - \frac{v^2}{\mu} \tag{7.18}$$

3. Compute: $e\cos E = t_1 = \frac{rv^2}{\mu} - 1$ and $e\sin E = t_2 = \frac{\|\vec{r}\|\left\|\dot{\vec{r}}\right\|}{\sqrt{\mu a}}$

4. Deduce the eccentricity e and the eccentric and mean anomalies E and M:

$$\begin{cases} e = \sqrt{t_1^2 + t_2^2} \\ E = \tan^{-1}\frac{t_2}{t_1} \\ M = E - e\sin E = E - t_2 \end{cases} \tag{7.19}$$

5. Compute for $i = 1, 2, 3$,
$$\begin{cases} P_i = \frac{r_i}{r}\cos E - \dot{r}_i\sqrt{\frac{a}{\mu}}\sin E \\ Q_i = \frac{\frac{r_i}{r}\sin E + \dot{r}_i\sqrt{\frac{a}{\mu}}(\cos E - e)}{\sqrt{1-e^2}} \end{cases}$$

6. Deduce the inclination i, the argument of perigee ω and the longitude of ascending node Ω:

$$\begin{cases} i = \tan^{-1}\frac{\sqrt{P_3^2+Q_3^2}}{P_1Q_2 - P_2Q_1} \\ \omega = \tan^{-1}\frac{P_3}{Q_3} \\ \Omega = \tan^{-1}\frac{P_2Q_3 - P_3Q_2}{P_1Q_3 - P_3Q_1} \end{cases} \tag{7.20}$$

7.1.5 Mathematical Analysis of the J_2-problem

This analysis is of no need for the numerical computations. However, it is essential for understanding the problem.
After reducing the problem to Lagrange equations (in the six orbital elements), a perturbation method is described for solving those equation (as detailed by Zarrouati in [63]). Then the results are interpreted for clarifying the effects of the J_2-Perturbation on the satellite motion, as given by Duriez in [65].

Reduction to Lagrange Equations:
Let us first split the J_2-Perturbed Earth's Gravitational Potential
$U = u(J_2) = \frac{\mu}{r}\left\{1 + \left(\frac{r_{eq}}{r}\right)^2 J_2 P_2(\sin\varphi)\right\}$ into two components:

$$U = \frac{\mu}{r} + R$$

where μ/r is the Newton component and R is the perturbing potential:

$$R = +\frac{\mu}{r}\left(\frac{r_{eq}}{r}\right)^2 J_2 P_2(\sin\varphi). \tag{7.21}$$

A classical transformation of $\ddot{\vec{r}} = \vec{\nabla}u(J_2)$ (7.11) leads to Lagrange's equations, as detailed by Zarrouati in [63], that express the time rate of change of the orbital elements $(a, e, i, \omega, \Omega, M)$ as functions of the partial derivatives of the perturbing potential R:

$$\begin{pmatrix} \frac{da}{dt} \\ \frac{de}{dt} \\ \frac{di}{dt} \\ \frac{d\Omega}{dt} \\ \frac{d\omega}{dt} \\ \frac{dM}{dt} - n \end{pmatrix} = M_L(a, e, i) \begin{pmatrix} \frac{\partial R}{\partial a} \\ \frac{\partial R}{\partial e} \\ \frac{\partial R}{\partial i} \\ \frac{\partial R}{\partial \Omega} \\ \frac{\partial R}{\partial \omega} \\ \frac{\partial R}{\partial M} \end{pmatrix} \tag{7.22}$$

where $n = \sqrt{\frac{\mu}{a^3}}$ is the mean osculating motion and $M_L(a, e, i)$, is a square 6-by-6 antisymmetric matrix depending on the 3 first orbital elements a, e, i and given by:

$$M_L(a,e,i) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & M_{16} \\ 0 & 0 & 0 & 0 & M_{25} & M_{26} \\ 0 & 0 & 0 & M_{34} & M_{35} & 0 \\ 0 & 0 & M_{43} & 0 & 0 & 0 \\ 0 & M_{52} & M_{53} & 0 & 0 & 0 \\ M_{61} & M_{62} & 0 & 0 & 0 & 0 \end{pmatrix}$$

with:

$M_{16} = -M_{61} = \frac{2}{na}$

$M_{26} = -M_{62} = \frac{1-e^2}{na^2e}$

$M_{52} = -M_{25} = \frac{\sqrt{1-e^2}}{na^2e}$

$M_{43} = -M_{34} = \frac{1}{na^2 \sin i\sqrt{1-e^2}}$

$M_{35} = -M_{53} = \frac{cosi}{na^2 \sin i\sqrt{1-e^2}}$

Thus, Lagrange's equations are explicitly expressed by:

$$\begin{cases} \frac{da}{dt} = \frac{2}{na}\frac{\partial R}{\partial M} \\ \frac{de}{dt} = -\frac{1-e^2}{na^2e}\left(\frac{\partial R}{\partial \omega} - \sqrt{1-e^2}\frac{\partial R}{\partial M}\right) \\ \frac{di}{dt} = -\frac{1}{na^2 \sin i\sqrt{1-e^2}}\left(\frac{\partial R}{\partial \Omega} - \cos i\frac{\partial R}{\partial \omega}\right) \\ \frac{d\Omega}{dt} = \frac{1}{na^2 \sin i\sqrt{1-e^2}}\frac{\partial R}{\partial i} \\ \frac{d\omega}{dt} = \frac{\sqrt{1-e^2}}{na^2e}\frac{\partial R}{\partial e} - \frac{\cos i}{na^2 \sin i\sqrt{1-e^2}}\frac{\partial R}{\partial i} \\ \frac{dM}{dt} = n - \frac{2}{na}\frac{\partial R}{\partial a} - \frac{1-e^2}{na^2e}\frac{\partial R}{\partial e} \end{cases} \tag{7.23}$$

Note that the time rate of change of the three first orbital elements involves only the partial derivatives of R with respect to the three last orbital elements, and vice-versa.
Note also that the presence of e and $\sin i$ in the denominator of these equations can cause some problem for quasi-circular orbits (small e) and for equatorial orbits (small i). A change of variables must then be used for getting a better conditioned system of equations, as was done in [63].

Perturbation Method for Solving Lagrange's Equations (as given in [63]):
Let $A = (a, e, i, \omega, \Omega, M - nt)$ be the vector of modified orbital elements.
Lagrange's equations can then be written as:

$$\frac{dA}{dt} = \epsilon D(A,t), \tag{7.24}$$

where $\epsilon = J_2$ and D is the differential operator such that $\epsilon D(A, t)$ describes the right-hand-sides of Lagrange's equations.
However, the expression $\epsilon D(A, t)$ is quite complicated and does not allow an analytic and direct integration of (7.24).

Note first that the differential system

$$\frac{dA}{dt} = 0, \tag{7.25}$$

describes the the motion of a Keplerian motion and can be solved exactly.

Moreover, and since $\epsilon = J_2 \approx 10^{-3}$, then $\epsilon D(A,t)$ is a small term of order J_2.
It follows that Lagrange's equations (7.24) can be formulated by adding a "small" term to the Keplerian problem (7.25) that is exactly solvable.

Hence, the perturbation theory is applicable and consists of finding an approximate solution to the J_2-problem (7.24) which cannot be solved exactly, by starting from the exact solution of the related Keplerian problem (7.25).
The perturbation method leads to an expression of the desired solution in terms of a power series in the "small" parameter $\epsilon = J_2$ that quantifies the deviation from the exactly solvable problem. The leading term in the power series is the solution of the exactly solvable problem, while further terms describe the deviation in the solution, due to the deviation from the initial problem.
Formally the form of the approximate full solution is:

$$A = \epsilon^0 A_0 + \epsilon^1 A_1 + \epsilon^2 A_2 + ...$$

where A_0 is the known solution to the exactly solvable initial problem and A_1,A_2,... represents the "higher orders" which are found iteratively by some systematic procedure. For small ϵ, these higher orders are presumed to become successively less important.

Usually, an approximate "perturbation" solution is obtained by truncating the series by keeping only the first two terms,

$$A \approx A_0 + \epsilon^1 A_1,$$

that is: the initial exact solution and the "first order" perturbation correction.

- *First iteration:*
  Only the centered component μ/r of the potential is considered. The resulting system (7.25) to integrate:
  $$\frac{dA}{dt} = 0$$
  has an exact solution that is known and can be expressed as:
  $$A_0 = (a, e, i, \omega, \Omega, M - nt) = \text{constants}$$

- *Second iteration:*
  The perturbing gravitational potential R is expressed in terms of the orbital elements:
  $$R = \frac{3\mu r_{eq}^2 J_2}{2a^3} \left(\frac{1 + e\cos\theta}{1 - e^2} \right)^3 \left[\frac{1}{3} + \frac{\sin^2 i}{2} \left(\cos 2(\theta + \omega) - 1 \right) \right],$$
  with $\frac{d\theta}{dM} = \frac{a^2\sqrt{1-e^2}}{r^2}$, for getting the new Lagrange system to integrate:
  $$\frac{dA}{dt} = J_2 D(A_0, t). \tag{7.26}$$
  The right hand side of this system depends only on constant parameters and on the true anomaly θ. Therefore, it varies periodically with the same period as that of the

Keplerian revolution. Thus, it can be decomposed as follows:

$$J_2 D(A_0, t) = J_2 D_c(A_0, t) + J_2 D_s(A_0),$$

where $J_2 D_s(A_0)$ is the average over one revolution of $J_2 D(A_0, t)$, and $J_2 D_c(A_0, t)$ represents the periodic deviation of $J_2 D(A_0, t)$ with respect to this average, with:

$$\begin{cases} D_s(A_0) = \frac{1}{2\pi} \int_0^{2\pi} D(A_0, t) dM \\ \frac{1}{2\pi} \int_0^{2\pi} D_c(A_0, t) dM = 0. \end{cases}$$

The integration of (7.26), with decomposed right hand side, yields:

$$A(t) = A_0 + J_2 \Delta A_c(t) + J_2 \Delta A_s(t), \tag{7.27}$$

where

$$\begin{cases} J_2 \Delta A_c(t) = \int_{t_0}^{t} J_2 D_c(A_0, t) dt \\ J_2 \Delta A_s(t) = \int_{t_0}^{t} J_2 D_s(A_0) dt \quad = J_2 D_s(A_0)(t - t_0). \end{cases}$$

Since the average of $D_c(A_0, t)$ is zero over one revolution, the term $J_2 \Delta A_c(t)$ represents periodic variations of the orbital elements (with the same orbital period), and is hence called "short-periodic" term.
As for the term $J_2 \Delta A_s(t)$, and since $D_s(A_0)$ is constant for one revolution, it represents linear in time variations of the orbital elements and is called "secular" term.

The explicit expressions for $J_2 \Delta A_s(t)$ are simple:

$$\begin{cases} \Delta a_s = 0 \\ \Delta e_s = 0 \\ \Delta i_s = 0 \\ \Delta \Omega_s = -\frac{3}{2} n J_2 \left(\frac{r_{eq}}{a}\right)^2 \frac{\cos i}{(1-e^2)^2} (t - t_0) \\ \Delta \omega_s = \frac{3}{4} n J_2 \left(\frac{r_{eq}}{a}\right)^2 \frac{\left(4 - 5\sin^2 i\right)}{(1-e^2)^2} (t - t_0) \\ \Delta M_s = \left[n + \frac{3}{4} n J_2 \left(\frac{r_{eq}}{a}\right)^2 \frac{\left(2 - 3\sin^2 i\right)}{(1-e^2)^3/2} \right] (t - t_0) \end{cases} \tag{7.28}$$

whereas the explicit expressions for $J_2 \Delta A_c(t)$ are rather complicated and can be found in [63]. Note that all of the six orbital elements undergo short-periodic variations (that is, on the period of the orbit).

- *Third iteration:*
  One should now integrate the new Lagrange system:

$$\frac{dA}{dt} = J_2 D(A_0 + J_2 \Delta A_c + J_2 \Delta A_s). \tag{7.29}$$

The solution to this system has the same form as (7.27) but with an additional term:

$$A(t) = A_0 + J_2 \Delta A_c(t) + J_2 \Delta A_s(t) + \Delta A_L. \tag{7.30}$$

Zarrouati gives an analysis of this additional term and proves in [63] that this term is of order J_2 (and not J_2^2 as expected) and is a harmonic function of $k\omega$. Since ω varies much slower than θ, ΔA_L is said to be a "long-periodic" term.

The explicit expressions for ΔA_L are given in [63] and we only cite here the remarkable result:

$$\Delta a_L = 0.$$

Hence, the semimajor axis does not change long-periodically, whereas all of the other orbital elements undergo long-period variations.
Note finally that if a solution of order J_2 is requires, one should take into account long-periodical terms due to J_3 and J_4 (that are of order J_2^2), since those terms appears in the third iteration when terms in J_2^2 are considered in Lagrange's equation and yield terms of order J_2.

Effects of the J_2-Perturbation on the Orbital Elements:
If the previous iterative method is restricted to the second iteration and *integrated over one revolution*, then only the secular terms will remain. It follows, as detailed by Duriez in [65], that the averaged orbital elements $\bar{a}$, $\bar{e}$ and $\bar{i}$ are constant (and equal to their initial values):

$$\bar{a} = a_0, \quad \bar{e} = e_0, \quad \bar{i} = i_0, \quad \text{and} \quad \bar{n} = n_0, \quad \text{with} \quad n_0^2 a_0^3 = \mu \tag{7.31}$$

whereas the averaged angular orbital elements $\bar{\Omega}$, $\bar{\omega}$ and $\bar{M}$ are linear functions of t:

$$\bar{\Omega} = \Omega_0 + n_\Omega t \quad \text{with} \quad n_\Omega = -\frac{3}{2} n_0 J_2 \frac{r_{eq}^2}{a_0^2} \frac{\cos i_0}{\left(1 - e_0^2\right)^2} \tag{7.32}$$

$$\bar{\omega} = \omega_0 + n_\omega t \quad \text{with} \quad n_\omega = \frac{3}{4} n_0 J_2 \frac{r_{eq}^2}{a_0^2} \frac{\left(4 - 5\sin^2 i_0\right)}{\left(1 - e_0^2\right)^2} \tag{7.33}$$

$$\bar{M} = M_0 + n_M t \quad \text{with} \quad n_M = n_0 + \frac{3}{4} n_0 J_2 \frac{r_{eq}^2}{a_0^2} \frac{\left(2 - 3\sin^2 i_0\right)}{\left(1 - e_0^2\right)^3 / 2} \tag{7.34}$$

n_Ω and n_ω being respectively the rate of change of the averaged longitude of the ascending node and that of the averaged argument of the perigee.

From the formulas above, Duriez derived in [65] the following conclusions:

- Since $J_2 \approx 10^{-3}$, one notes that the angular velocities n_Ω and n_ω of the ascending node and the perigee are around a thousand times smaller than the constant mean angular velocity n_0 of the satellite on its orbit (ex: if a satellite performs 15 revolutions per day, then the ascending node and the perigee rotate about 5° per day). *Therefore, the ascending node and the perigee also rotate around the earth, but much slower than the satellite itself.*

- For fixed values of e_0 and i_0, the velocities n_Ω and n_ω decrease very quickly when a_0 increases: they vary like $\frac{n_0}{a_0^2}$ .Thus, for two orbits of semi-major axes a_0 and $a_0^{'}$, one has $n_0^2 a_0^3 = \mu = n_0^{'2} a_0^{'3}$ yielding

$$\frac{n_0^{'}}{a_0^{'2}} = \frac{n_0}{a_0^2} \left(\frac{a_0}{a_0^{'}}\right)^{7/2}.$$

Doubling for example the semi-major axis results in dividing the angular velocities n_Ω and n_ω of the ascending node and the perigee by $2^{7/2} \approx 11.3$.
Thus, the effect of the flattening of the earth is of greater importance for the satellites that are closer to earth.

- For fixed values of e_0 and i_0, n_Ω and n_ω vary inversely with the semi-latus rectum of the ellipse $p_0 = a_0(1 - e_0^2)$.

- The J_2-Perturbation causes the orbit plane to precess like a top, the precession frequency being n_Ω. Note that the *precession of the ascending node* depends on the sign of n_Ω (that is the sign of $\cos i_0$). Thus, the precession of the ascending node is maximum for small values of i_0 that are close to 0, is retrograde when $i_0 < 90°$ and advances when $i_0 > 90°$.

- Similarly, the *relative variation of the perigee with respect to the ascending node* depends on the sign of n_ω, that is the sign of $\left(4 - 5\sin^2 i_0\right)$, which is 0 for $i_1 = 63°26'$ and for $i_2 = 180° - i_1 = 116°34'$. As a result, the perigee advances relative to the ascending node if $i_0 < i_1$ or $i_0 > i_2$, is invariant if $i_0 = i_1$ or $i_0 = i_2$ (critical inclinations) and is retrograde if $i_1 < i_0 < i_2$.
  Note that when the inclination i_0 is close to the critical values i_1 and i_2, the previous approximation is no more valid since n_ω might then be of the order of J_2^2 making the term in J_4 no more negligible.

- The mean motion n_M is equal to n_0 when $\left(2 - 3\sin^2 i_0\right) = 0$, that is when $i_0 = 54°44'$ or $125°16'$, and is larger or less than n_0 depending on if the inclination i_0 is between these two values or not. However, and since the mean motion n_M is never zero, there is no critical inclinations in this case.

7.2 J_2-perturbed Motion as a System of ODE's

In the next sections of this chapter, we solve the J_2-perturbed problem numerically, and in a time-parallel way.
But first, we define the system of rectangular coordinates that we will use and reduce the J_2-perturbed problem to a first order system of ODE's of the general form (S).

7.2.1 Our Coordinate System: Initially Perifocal Coordinate Frame (IPQW-Frame)

In the perturbed motion, the trajectory of the satellite is always tangential to an instantaneous ellipse (the osculating ellipse) defined by the instantaneous values of the time-varying orbital elements.
Therefore, this trajectory is in a moving orbital plane (containing the osculating ellipse), defined instantaneously by $\left(\vec{r}(t), \dot{\vec{r}}(t)\right)$, so that the PQW-Frame is a moving frame in which $\vec{P}$ and $\vec{Q}$ lie in the instantaneous orbital plane, while $\vec{W}$ is normal to it.

Since the moving PQ-plane expresses the perturbation about what would have been the Keplerian plane in a Keplerian motion (i.e. the plane containing the osculating ellipse at the initial time t_0), we will adopt in our computations the **IPQW-frame**, which is the **PQW-frame corresponding to the initial conditions** and keep it unchanged with the time t. We consider that the effect of the perturbation will be more significant in this frame than in any other.

7.2.2 Explicit Expression of $\vec{\nabla} U$

Explicit Expression of U:
Let $\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$, implying $\dot{\vec{r}} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix}$ and $\ddot{\vec{r}} = \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix}$.
The considered gravitational potential is given by (7.7):

$$U = u(J_2) = \frac{\mu}{r}\left\{1 + \left(\frac{r_{eq}}{r}\right)^2 J_2 P_2(\sin\varphi)\right\}.$$

First of all, it is interesting to notice that the gravitational potential $U = u(J_2)$ is the sum of a term due to the centered component of the gravity of the earth (that we will call "Keplerian term", since it is the only term in a Keplerian motion) and a second term due to the J_2-perturbation:

$$U = U_K + U_P, \tag{7.35}$$

where:

$$U_K = \frac{\mu}{r}, \tag{7.36}$$

and:

$$U_P = \frac{\mu}{r}\left(\frac{r_{eq}}{r}\right)^2 J_2 P_2(\sin\varphi). \tag{7.37}$$

Recall that $r = \|\vec{r}\|_2 = \sqrt{x^2 + y^2 + z^2}$ and μ, r_{eq} and J_2 are constants given in (refmu), (refreq) and (refJ2) respectively.

For getting an explicit expression of U_P (in terms of (x, y, z)), one must first express

the geocentric latitude φ in terms of the rectangular coordinates. This latitude is the angle that the ray, from the center of the Earth to the satellite, makes with the plane of the equator (North latitude is positive, South latitude is negative).
It follows that $\sin\varphi$ expresses very simply in the ECI-frame, since its xy-plane is the equatorial plane, as:

$$\sin\varphi = \left[\frac{z}{r}\right]_{ECI},$$

yielding, in the ECI-frame:

$$P_2(\sin\varphi) = \frac{3}{2}\sin^2\varphi - \frac{1}{2} = \left[\frac{3}{2}\frac{z^2}{r^2} - \frac{1}{2}\right]_{ECI}.$$

Therefore:

$$U_P = \left[\frac{3\mu J_2 r_{eq}^2}{2}\frac{z^2}{r^5} - \frac{\mu J_2 r_{eq}^2}{2}\frac{1}{r^3}\right]_{ECI}. \tag{7.38}$$

Explicit Expression of $\vec{\nabla}U$:
Note first that:

$$\vec{\nabla}U = \vec{\nabla}U_K + \vec{\nabla}U_P,$$

and let:

$$\vec{f_K}(\vec{r}) = \vec{\nabla}U_K$$

and:

$$\vec{f_P}(\vec{r}) = \vec{\nabla}U_P,$$

making:

$$\vec{\nabla}U = \vec{f_K}(\vec{r}) + \vec{f_P}(\vec{r}), \tag{7.39}$$

In any rectangular coordinate system in which $\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ and $r = \sqrt{x^2+y^2+z^2}$, the partial differentiation of U_K yields:

$$\vec{f_K}(\vec{r}) = \vec{f_K}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{\partial U_K}{\partial x} \\ \frac{\partial U_K}{\partial y} \\ \frac{\partial U_K}{\partial z} \end{pmatrix} = \begin{pmatrix} -\mu\frac{x}{r^3} \\ -\mu\frac{y}{r^3} \\ -\mu\frac{z}{r^3} \end{pmatrix}. \tag{7.40}$$

But for the partial differentiation of U_P to be explicitly carried out, it should be done in the ECI-frame, where one has the explicit expression (7.38) for U_P:

$$\left[\vec{f_P}\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{ECI}\right]_{ECI} = \begin{pmatrix} \frac{\partial U_P}{\partial x} \\ \frac{\partial U_P}{\partial y} \\ \frac{\partial U_P}{\partial z} \end{pmatrix}_{ECI} = \begin{pmatrix} \frac{3\mu J_2 r_{eq}^2}{2}\left(1-5\frac{z^2}{r^2}\right)\frac{x}{r^5} \\ \frac{3\mu J_2 r_{eq}^2}{2}\left(1-5\frac{z^2}{r^2}\right)\frac{y}{r^5} \\ \frac{3\mu J_2 r_{eq}^2}{2}\left(3-5\frac{z^2}{r^2}\right)\frac{z}{r^5} \end{pmatrix}_{ECI} \tag{7.41}$$

Using the transformation matrix A given in (7.14) and formulae (7.15) and (7.16), one deduces:

$$\left[\vec{f_P}\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{IPQW}\right]_{IPQW} = A\left[\vec{f_P}\left(A^T\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{IPQW}\right)\right]_{ECI}, \tag{7.42}$$

where $\left[\vec{f_P}(.)_{ECI}\right]_{ECI}$ is given by (7.41).

7.2.3 Equivalent System of First Order ODE's

The differential equation (7.11), combined with initial conditions, yields a second-order initial value problem, describing the J_2-perturbed problem, in which one seeks $\vec{r} : [0, \infty) \to \mathbb{R}^3$ such that:

$$\begin{cases} \ddot{\vec{r}}(t) = \vec{f}(\vec{r}), \quad t > 0, & (7.43.1) \\ \vec{r}(0) = \vec{r_0}, & (7.43.2) \\ \dot{\vec{r}}(0) = \vec{\dot{r}_0}, & (7.43.3) \end{cases} \tag{7.43}$$

where:

$$\vec{f}(\vec{r}) = \vec{\nabla} U. \tag{7.44}$$

Lowering the Order:
Letting:

$$\vec{r_1}(t) = \vec{r}(t) \qquad \text{and} \qquad \vec{r_2}(t) = \dot{\vec{r}}(t).$$

makes solving the second order problem (7.43) equivalent to solving, for $\vec{r_1}, \vec{r_2} : [0, \infty) \to \mathbb{R}^3$, the first order problem:

$$\begin{cases} \frac{d\vec{r_1}}{dt} = \vec{r_2}, \quad t > 0, & (7.45.1) \\ \frac{d\vec{r_2}}{dt} = \vec{f}(\vec{r_1}), \quad t > 0, & (7.45.2) \\ \vec{r_1}(0) = \vec{r_0}, & (7.45.3) \\ \vec{r_2}(0) = \vec{\dot{r}_0}. & (7.45.4) \end{cases} \tag{7.45}$$

We let now:
$Y = \begin{pmatrix} \vec{r_1} \\ \vec{r_2} \end{pmatrix} \in \mathbb{R}^6, \quad Y_0 = \begin{pmatrix} \vec{r_1}(0) \\ \vec{r_2}(0) \end{pmatrix} = \begin{pmatrix} \vec{r_0} \\ \vec{\dot{r}_0} \end{pmatrix} \in \mathbb{R}^6.$

Problem (7.45) can then be rewritten, in a more compact way, as a first-order initial value problem of dimension 6, of the general form (S), in which one seeks a function $Y : [0, \infty) \to \mathbb{R}^6$ satisfying:

$$\begin{cases} \frac{dY}{dt} = F(Y), \quad t > 0, \\ Y(0) = Y_0 \end{cases} \tag{7.46}$$

where:

$$F(Y) = \begin{pmatrix} \vec{r_2} \\ \vec{f}(\vec{r_1}) \end{pmatrix} \in \mathbb{R}^6.$$

Since $\vec{f}(\vec{r_1}) = \vec{\nabla} U = \vec{f_K}(\vec{r}) + \vec{f_P}(\vec{r})$, as stated in (7.39), it follows that $F(Y)$ can also be split in a sum of a Keplerian term and a perturbing term:

$$F(Y) = F_K(Y) + F_P(Y), \tag{7.47}$$

with:

$$F_K(Y) = \begin{pmatrix} \vec{r_2} \\ \vec{f_K}(\vec{r_1}) \end{pmatrix}, \tag{7.48}$$

$$F_P(Y) = \begin{pmatrix} \vec{0} \\ \vec{f_P}(\vec{r_1}) \end{pmatrix}. \tag{7.49}$$

Expression of $F(Y)$, in the IPQW-frame:

$$[\vec{r}]_{IPQW} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \text{ and } \left[\dot{\vec{r}}\right]_{IPQW} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} \text{ yield: } [Y]_{IPQW} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \\ Y_6 \end{pmatrix} = \begin{pmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix}.$$

The expressions of $\vec{f_K}(.)$ and $\vec{f_P}(.)$ are given relatively to the coordinates in the IPQW-frame, in (7.40) and (7.42) respectively, and yield:

$$[F_K(Y)]_{IPQW} = \begin{pmatrix} Y_4 \\ Y_5 \\ Y_6 \\ -\mu\frac{Y_1}{R^3} \\ -\mu\frac{Y_2}{R^3} \\ -\mu\frac{Y_3}{R^3} \end{pmatrix} \tag{7.50}$$

and:

$$[F_P(Y)]_{IPQW} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ A\vec{f_P}\left[A^T \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix}\right] \end{pmatrix}, \tag{7.51}$$

where $\vec{f_P}(.)$ is given in (7.42) and $R = \sqrt{Y_1^2 + Y_2^2 + Y_3^2}$.

7.3 RaTI Applied to a Keplerian Motion

7.3.1 EOS Condition, Rescaling & Invariance

In a Keplerian motion, as described in the first section of this chapter, the satellite moves in a fixed plane, that is completely defined by the initial position and velocity vectors, along a fixed ellipse having the center of the earth as one focus. This confines the satellite's trajectory in the xy-plane of our IPQW- frame, making $Y_3(t) = Y_6(t) = 0 \ (\forall t)$.
Besides, a Kelplerian motion is periodic of period $P = 2\pi\sqrt{\frac{a^3}{\mu}}$.
Problem (7.46) reduces, in such a case, to:

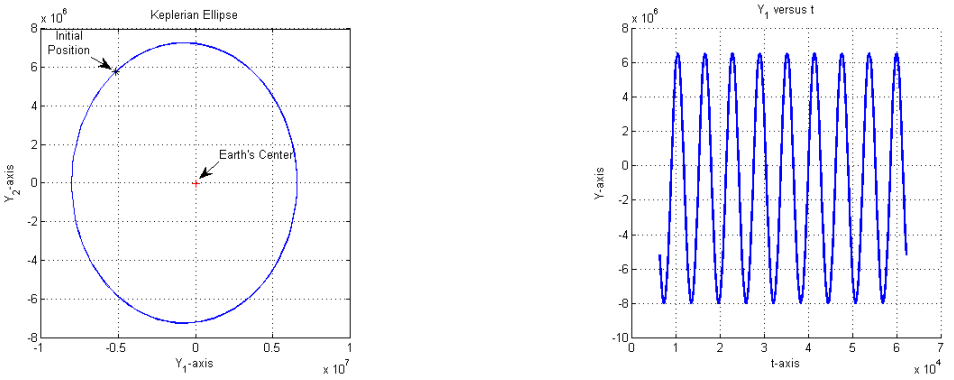
$$\begin{cases} \frac{dY}{dt} = F_K(Y), & t > 0 \\ Y(0) = Y_0, \end{cases} \tag{7.52}$$

where:

$$F_K(Y) = \begin{pmatrix} Y_4 \\ Y_5 \\ 0 \\ -\mu\frac{Y_1}{R^3} \\ -\mu\frac{Y_2}{R^3} \\ 0 \end{pmatrix}. \tag{7.53}$$

Selection of an EOS Condition
In addition to yielding $Y_3(t) = Y_6(t) = 0$ for all t, a Keplerian motion results in periodic

oscillations of $Y_i(t)_{i\in\{1,2,4,5\}}$ with respect to the time t.
Figure 7.3.1 illustrates such behavior.



(a) Keplerian Ellipse

(b) Periodic Solution: Y_1 versus t

Figure 7.5: Satellite - Keplerian motion ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$).

As discussed in Chapter 2, in case of oscillatory behavior of the solution, the functions $\{E_n\}$ governing the EOS condition could be that defined in (2.21), for all $n > 0$, by:

$$\forall Y(t) \in \mathbb{R}^k, \qquad E_n\left[Y(t)\right] = \theta_n\left[Y(t)\right] - 2\pi,$$

where $\theta_n(W)$ is the rotation angle (in the xy-plane), at each instant t, given in (2.17) and evaluated using the procedure described in definition 7 of Chapter 2.
The n^{th} slice is then ended when the solution completes a period in the Keplerian plane, making the polar angle, at the end of every slice, equal to the initial polar angle corresponding to the initial value Y_0.
This yields the following EOS condition:

$$\begin{array}{c} \text{at } t = T_n,\ E_n\left[Y\left(T_n\right)\right] = 0, \quad \Longleftrightarrow \quad \theta_n\left[Y(T_n)\right] = 2\pi, \\ \text{with} \quad \theta_n\left[Y(t)\right] < 2\pi, \ \text{ if } T_{n-1} \leq t < T_n. \end{array} \tag{7.54}$$

Figure 7.6 represents the first component of the solution Y as a function of t, on the first 10 slices, for the same initial conditions than those of figure 7.3.1.

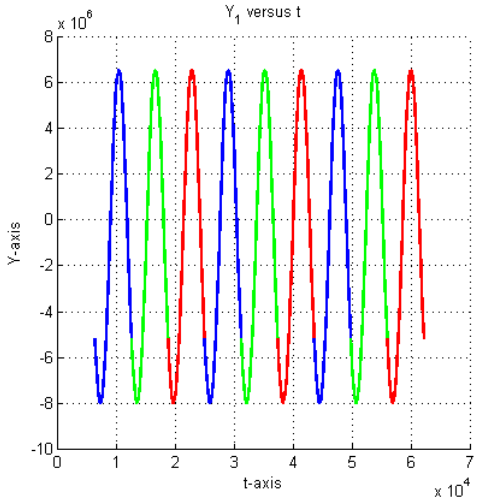


Figure 7.6: Satellite - EOS Condition for Keplerian Motion ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$)

Problem (7.52) is then equivalent to the sequence of initial value shooting problems in which one seeks, on each n^{th} slice ($n \geq 1$), for the time T_n and the solution Y such that:

$$\left\{ \begin{array}{ll} \frac{dY}{dt} = F_K(Y) & t > T_{n-1}, \quad (7.55.1) \\ Y(T_{n-1}) = Y_{n-1}, & (7.55.2) \\ \theta_n\left[Y(T_n)\right] = 2\pi \text{ and } \forall t \in (T_{n-1}, T_n), \;\; \theta_n\left[Y(t)\right] < 2\pi, & (7.55.3) \end{array} \right. \quad (7.55)$$

where $F_K(Y)$ is given in (7.53).

Change of Variables:
Note first that the periodicity of the motion inflicts to keep the time-scale invariant, on all slices and therefore, to choose a time-rescaling factor $\beta_n = 1$ $(\forall n)$. The only effect of the change of variable $t = T_{n-1} + \beta_n s$, is then to reset the time-variable to zero, at the beginning of each slice:

$$t = T_{n-1} + s. \quad (7.56)$$

Knowing that $Y_3(t) = Y_6(t) = 0$ for all $t \geq 0$ and assuming, without loss of generality, that $Y_{n,i} \neq 0$ for all $i \in \{1, 2, 4, 5\}$ and for all $n > 1$, one has, according to (1.6):

$$\forall n > 1, \;\; \alpha_n = \begin{pmatrix} Y_{n-1,1} \\ Y_{n-1,2} \\ 1 \\ Y_{n-1,4} \\ Y_{n-1,5} \\ 1 \end{pmatrix}. \quad (7.57)$$

Recall that the general change of the solution variable Y is given in (1.5.2) by: $Y(t) = Y_{n-1} + D_n Z_n(s)$, where D_n is the diagonal matrix associated with the vector α_n. In this case, this change of variable yields:

$$\left\{ \begin{array}{ll} Y_i(t) = Y_{n-1,i}(1 + Z_{n,i}(s)), & \text{if } i \in \{1, 2, 4, 5\}, \\ Y_i(t) = Z_{n,i}(s) = 0, & \text{if } i \in \{3, 6\}. \end{array} \right. \quad (7.58)$$

Rescaled EOS condition:

As shown in (2.27), the functions $\{E_n\}$ governing the EOS condition (7.54) rescale to a unique function $H : \mathbb{R}^6 \to \mathbb{R}$ defined, independently of n, by:

$$\forall Z_n(s) \in \mathbb{R}^k,\ H\left[Z_n(s)\right] = \tilde{\theta}\left[Z_n(s)\right] - 2\pi,$$

where $\tilde{\theta}$ is given in (2.26) by: $\tilde{\theta}\left[Z_n(s)\right] = \theta\left[Y_{n-1} + D_n Z_n(s)\right]$.
The n^{th} slice is ended as soon as the rescaled solution $Z(.)$ satisfies:

$$\begin{array}{cccc} \text{at } s = s_n, & H\left[Z_n(s_n)\right] = 0 & \Longleftrightarrow & \tilde{\theta}\left[Z_n(s_n)\right] = 2\pi, \\ \text{with} & \tilde{\theta}\left[Z_n(s)\right] < 2\pi, & \text{if } 0 < s < s_n. & \end{array} \tag{7.59}$$

Resulting Rescaled Systems:
From the change of variables (7.56) and (7.58) follows the vector-wise differentiation:
$\frac{dZ_n}{ds} = \frac{dZ_n}{dY}\frac{dY}{dt}\frac{dt}{ds} = D_n^{-1}\frac{dY}{dt}\beta_n = D_n^{-1}\frac{dY}{dt} = D_n^{-1}F_K(Y)$,
where $F_K(Y)$ is given by (7.53). It implies:

$$G_{n,K}(Z_n) = \frac{dZ_n}{ds} = D_n^{-1}\begin{pmatrix} Y_4 \\ Y_5 \\ 0 \\ -\mu\frac{Y_1}{R^3} \\ -\mu\frac{Y_2}{R^3} \\ 0 \end{pmatrix} = \begin{pmatrix} \frac{Y_{n-1,4}}{Y_{n-1,1}}\left[1 + Z_{n,4}(s)\right] \\ \frac{Y_{n-1,5}}{Y_{n-1,2}}\left[1 + Z_{n,5}(s)\right] \\ 0 \\ -\mu\left(\frac{Y_{n-1,1}}{Y_{n-1,4}}\right)\frac{1+Z_{n,1}(s)}{R^3} \\ -\mu\left(\frac{Y_{n-1,2}}{Y_{n-1,5}}\right)\frac{1+Z_{n,2}(s)}{R^3} \\ 0 \end{pmatrix} \tag{7.60}$$

where $R = \sqrt{Y_1^2 + Y_2^2} = \sqrt{Y_{n-1,1}^2\left[1 + Z_{n,1}(s)\right]^2 + Y_{n-1,2}^2\left[1 + Z_{n,2}(s)\right]^2}$.

Using the End-Of-Slice condition (7.54) and rescaling with the change of variables (7.56) and (7.58), make the satellite problem (7.68) equivalent to the following sequence of rescaled systems:

$$\left\{\begin{array}{ll} \frac{dZ_n}{ds} = G_{n,K}(Z_n),\ s > 0 & (7.61.1) \\ Z_n(0) = 0, & (7.61.2) \\ \tilde{\theta}\left[Z_n(s_n)\right] = 2\pi, \quad \text{and} \quad \forall s \in (0, s_n),\ \tilde{\theta}\left[Z_n(s)\right] < 2\pi. & (7.61.3) \end{array}\right. \tag{7.61}$$

where $G_{n,K}(Z_n)$ is given in (7.60).

Invariance:
Knowing the periodicity of the motion, and since the EOS condition is chosen so as to end a slice at the end of every period, one has then:

$$\forall n,\ \forall i, \quad Y_{n,i} = Y_{0,i}. \tag{7.62}$$

This makes:

$$\forall n, \quad G_{n,K}(Z_n) = \begin{pmatrix} \frac{Y_{0,4}}{Y_{0,1}}\left[1+Z_{n,4}(s)\right] \\ \frac{Y_{0,5}}{Y_{0,2}}\left[1+Z_{n,5}(s)\right] \\ 0 \\ -\mu\left(\frac{Y_{0,1}}{Y_{0,4}}\right)\frac{1+Z_{n,1}(s)}{R^3} \\ -\mu\left(\frac{Y_{0,2}}{Y_{0,5}}\right)\frac{1+Z_{n,2}(s)}{R^3} \\ 0 \end{pmatrix} = G_{1,K}(Z_n). \tag{7.63}$$

where $R = \sqrt{Y_1^2 + Y_2^2} = \sqrt{Y_{0,1}^2\left[1+Z_{n,1}(s)\right]^2 + Y_{0,2}^2\left[1+Z_{n,2}(s)\right]^2}$.
Therefore: $\forall n,\ G_n(.) = G_1(.)$. Together with the invariance of the function H governing the rescaled EOS condition (7.59), this yields the invariance of the rescaled systems (7.61), according to definition 2 of Chapter 1.

7.3.2 Numerical Results: Application of RaTI

According to property 1, given in Chapter 1, the invariance of the rescaled problems (S'_n) implies:

$$\forall n > 1, \quad Z_n(.) = Z_1(.) \qquad \text{and} \qquad s_n = s_1,$$

(and therefore, a constant sequence $\{Y_n\}$: $\forall n,\ Y_n = Y_0$).
It follows that the rescaled solution Z_1 of the first time-slice solves all the slices, by simply applying, for all $n > 1$, the change of variables (7.56) and (7.58):

$$\forall n > 1, \quad \forall s \in [0, s_1], \quad \begin{cases} t = T_{n-1} + s, \\ Y_i(t) = Y_{0,i}(1 + Z_{1,i}(s)) & \text{if } i \in \{1,2,4,5\}, \\ Y_i(t) = Z_{1,i}(s) = 0 & \text{if } i \in \{3,6\}, \end{cases}$$

The numerical solution of this periodic problem can then be obtained, using the sequential Ratio-based Time Integration (RaTI) algorithm described in Chapter 4, for invariant cases.

Tables (7.1) and (7.2) below summarize the numerical results, for 12 different initial conditions. Each initial condition is given by its six Keplerian orbital elements $(a_0, e_0, i_0, \omega_0, \Omega_0, M_0)$.

This is done on one processor and the time of computation is compared to the time needed for solving sequentially the same number of slices, without using the invariance property.

In all the experiments:
- The time-step of integration that has been used is $\tau = 60$ seconds.
- The tolerance up to which is reached the EOS condition is $\epsilon_{tol}^{eos} = 10^{-11}$.
- The total number of slices is $N = 600$.

Case	1	2	3	4	5	6
e_0	0.1	0.1	0.1	0.0005	0.15	0.1
a_0	7300 km	7650 km	8000 km	7300 km	7300 km	7300 km
i_0	98°	98°	98°	98°	98°	80°
ω_0	10°	10°	10°	10°	10°	10°
Ω_0	45°	45°	45°	45°	45°	45°
M_0	123°	123°	123°	123°	123°	123°
n_{days}	43.11	46.24	49.45	43.11	43.11	43.11
T_{seq}	**28.64**	**32.21**	**32.7**	**27.67**	**29.37**	**28.69**
T_{inv}	**0.53**	**0.59**	**0.56**	**0.52**	**0.55**	**0.48**
Speed-up	**54.04**	**54.59**	**58.39**	**53.21**	**53.40**	**59.77**

Table 7.1: Satellite, Keplerian Motion - Application of RaTI (1).

Case	7	8	9	10	11	12
e_0	0.1	0.1	0.1	0.1	0.1	0.1
a_0	7300 km	7300 km	7300 km	7300 km	7300 km	7300 km
i_0	98°	98°	98°	98°	98°	98°
ω_0	5°	20°	10°	10°	10°	10°
Ω_0	45°	45°	10°	120°	45°	45°
M_0	123°	123°	123°	123°	20°	60°
n_{days}	43.11	43.11	43.11	43.11	43.11	43.11
T_{seq}	**29.1**	**29.04**	**28.72**	**28.69**	**29.52**	**29.51**
T_{inv}	**0.51**	**0.54**	**0.53**	**0.51**	**0.52**	**0.5**
Speed-up	**57.06**	**53.78**	**54.19**	**56.25**	**56.77**	**59.02**

Table 7.2: Satellite, Keplerian Motion - Application of RaTI (2).

Clearly, the application of RaTI algorithm results in a tremendous, but not surprising, speed-up of the computations.

7.4 RaPTI Applied to a J_2-Perturbed Motion

7.4.1 Choice of an EOS Condition

As stated in a previous section, the satellite orbit is, in this case of J_2-perturbed motion, an osculating ellipse of which the parameters vary at each instant. Since all of the instantaneous ellipses have the center of earth at one focus, the satellite orbit will pass regularly across the xy-plane (which is, in our case, the Keplerian plane corresponding to the initial conditions).
Besides, ans as discussed in Chapter 2, in case of oscillatory behavior of the solution and when one seeks a time-parallelism, the slicing technique would inflict that each slice is ended when the solution completes a full (or almost full) rotation.
One possible way to do it is to end each n^{th} slice $[T_{n-1}, T_n]$ whenever the satellite crosses the xy-plane, after completing a whole rotation, that is as soon as:

$$Y_{n,3} = 0, \tag{7.64}$$

for the second time, after having been satisfied by $Y_{n-1,3} = 0$ at the previous slice.
This EOS condition is governed by a family of functions $E_n : \mathbb{R}^6 \to \mathbb{R}$, given by:

$$\forall W \in \mathbb{R}^6, \quad E_n[W] = W_3 \tag{7.65}$$

and defined by adding, to the condition $E_n[Y_n] = 0$, the additional constraint that omits to stop when this condition is satisfied for the first time. This is intended to make the solution complete an almost full rotation in the phase plane.
Let L be the logical function defined by:

$$\begin{cases} L[Y(t)] = 0 & \text{if } Y_3(t) = 0 \quad \text{for the second time after } Y_3(T_{n-1}) = 0 \\ L[Y(t)] = 1 & \text{if } Y_3(t) \neq 0 \quad \text{or } Y_3(t) = 0 \text{ for the first time after } Y_3(T_{n-1}) = 0 \end{cases} \tag{7.66}$$

The EOS condition (7.64) translates then to:

$$L[Y(T_n)] = 0 \qquad \text{and} \qquad \forall t \in (T_{n-1}, T_n), \;\; L[Y(t)] \neq 0. \tag{7.67}$$

The behavior of the solution makes such EOS condition guaranteed to be reached (see figures 7.7 and 7.8)

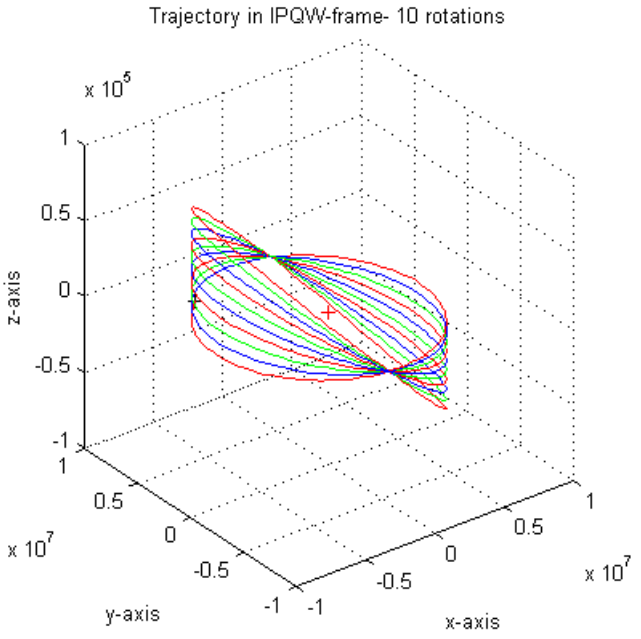


Figure 7.7: Satellite, J2-perturbed Motion - EOS Condition and Orbit ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$ and $M_0 = 123°$)

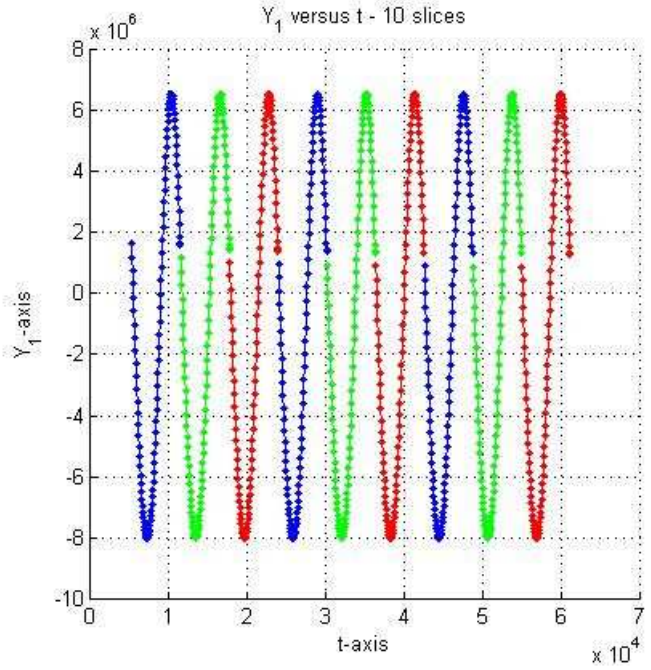


Figure 7.8: Satellite, J2-perturbed Motion - Y_1 versus t ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$)

Problem (7.46) is then equivalent to the sequence of initial value shooting problems in which one seeks, on each n^{th} slice ($n \geq 1$), for the time T_n and the solution $Y : [T_{n-1}, T_n] \longrightarrow \mathbb{R}^6$ such that:

$$\begin{cases} \frac{dY}{dt} = F(Y) = F_K(Y) + F_P(Y), \qquad t > T_{n-1}, & (7.68.1) \\ Y(T_{n-1}) = Y_{n-1}, & (7.68.2) \\ L[Y(T_n)] = 0 \quad \text{and} \quad \forall t \in (T_{n-1}, T_n), \;\; L[Y(t)] \neq 0, & (7.68.5) \end{cases} \tag{7.68}$$

where $F_K(Y)$ and $F_P(Y)$ are given by (7.48) and (7.49).

Remark 12 *Another natural choice of an EOS condition could have been to end a slice whenever the projection of the satellite's position on the xy-plane returns the polar angle to its initial value (modulo 2π). However, the numerical experiments gave a better ratio property with EOS condition (7.67).*

7.4.2 Rescaling

Change of Variables:
As described in formulae (1.5) of Chapter 1, the time t and the solution $Y(t)$ are changed into rescaled time s and rescaled solution $Z(s)$ respectively, as follows:

$$\begin{cases} t = T_{n-1} + \beta_n s, \\ Y(t) = Y_{n-1} + D_n Z_n(s). \end{cases}$$

Recall that $D_n = \text{diag}(\alpha_n) \in \mathbb{R}^{6\times 6}$ is a diagonal matrix associated with the vector $\alpha_n \in \mathbb{R}^6$ that is defined in (1.6), in terms of Y_{n-1} as:

$$\alpha_{n,i} = \begin{cases} Y_{n-1,i} & \text{if } Y_{n-1,i} \neq 0 \\ 1 & \text{if } Y_{n-1,i} = 0 \end{cases}$$

EOS condition (7.67) makes the third component of all starting values equal to 0, i.e.: $\forall n \geq 1,\ Y_3(T_n) = 0$.
Assuming also, without loss of generality, that:
* $Y_i(0) \neq 0$, for all $i \in \{1, 2, 4, 5\}$, and
* $Y_i(T_n) \neq 0$ for all $i \in \{1, 2, 4, 5, 6\}$ and for all $n \geq 1$, we get, in this non-Keplerian motion, the following expression for the vector α_n:

$$\forall n > 1,\ \ \alpha_{n,i} = \begin{cases} Y_{n-1,i} & \text{if } i \neq 3, \\ 1 & \text{if } i = 3, \end{cases} \tag{7.69}$$

yielding the change of variables:

$$\begin{cases} t = T_{n-1} + \beta_n s, & (7.70.1) \\ Y_i(t) = \begin{cases} Y_{n-1,i}(1 + Z_{n,i}(s)) & \text{if } i \neq 3 \\ Z_{n,i}(s) & \text{if } i = 3 \end{cases} & (7.70.2) \end{cases} \tag{7.70}$$

Note that for $n = 1$, and since $Y_{0,6} = 0$, we have also: $\alpha_{n,6} = 1$ yielding $Y_6(t) = Z_{n,6}(s)$. We will give, in all what follows, expressions corresponding to $n > 1$, knowing that the case $n = 1$ can easily be deduced.

Rescaled EOS Condition:
The equality involved in the EOS condition (7.64) rescales simply as:

$$Z_{n,3}(s_n)) = 0.$$

Let L_z be the logical function, derived from the function L given in (7.66) and defined by:

$$\begin{cases} L_z\left[Z_n(s)\right] = 0 & \text{if } Z_{n,3}(s)) = 0 \quad \text{for the second time after } Z_{n,3}(0)) = 0 \\ L_z\left[Z_n(s)\right] = 1 & \text{if } Z_{n,3}(s)) \neq 0 \quad \text{or } Z_{n,3}(s)) = 0 \text{ for the first time after } Z_{n,3}(0)) = 0 \end{cases} \tag{7.71}$$

The EOS condition (7.67) translates then to:

$$L_z\left[Z_n(s_n)\right] = 0 \qquad \text{and} \qquad \forall s \in (0, s_n),\ \ L_z\left[Z_n(s)\right] \neq 0. \tag{7.72}$$

Choice of the Time-Rescaling Factor β_n:
Since the global behavior of the solution consists of slight successive variations, that can be viewed as small perturbations of a periodic solution, we will keep the time-scale invariant by choosing:

$$\beta_n = 1 \tag{7.73}$$

The only effect of the change of variable (7.70.2) $t = T_{n-1} + \beta_n s = T_{n-1} + s$, is then to reset the time-variable to zero at the beginning of each slice, thus allowing to use a local time for solving the slices in parallel.

Resulting Rescaled Systems:
From the change of variables (7.70) and the choice (7.73) of β_n, follows the vector-wise differentiation:
$\frac{dZ_n}{ds} = \frac{dZ_n}{dY}\frac{dY}{dt}\frac{dt}{ds} = D_n^{-1}\frac{dY}{dt}\beta_n = D_n^{-1}\frac{dY}{dt} = D_n^{-1}F(Y) = D_n^{-1}\left[F_K(Y) + F_P(Y)\right]$.
where $F_K(Y)$ and $F_P(Y)$ are given by (7.50) and (7.51) respectively.
It implies:

$$\frac{dZ_n}{ds} = G_{n,K}(Z_n) + G_{n,P}(Z_n),$$

where $G_{n,K}(Z_n) = D_n^{-1} F_K(Y)$ and $G_{n,P}(Z_n) = D_n^{-1} F_P(Y)$.
Explicit expressions of $G_{n,K}(Z_n)$ and $G_{n,P}(Z_n)$ can be deduced from that of $F_K(Y)$ and $F_P(Y)$, as described below.

$$G_{n,K}(Z_n) = D_n^{-1} F_K(Y) = D_n^{-1} \begin{pmatrix} Y_4 \\ Y_5 \\ Y_6 \\ -\mu \frac{Y_1}{R^3} \\ -\mu \frac{Y_2}{R^3} \\ -\mu \frac{Y_3}{R^3} \end{pmatrix} = \begin{pmatrix} \frac{Y_{n-1,4}}{Y_{n-1,1}} [1 + Z_{n,4}(s)] \\ \frac{Y_{n-1,5}}{Y_{n-1,2}} [1 + Z_{n,5}(s)] \\ Y_{n-1,6} [1 + Z_{n,6}(s)] \\ -\mu \left(\frac{Y_{n-1,1}}{Y_{n-1,4}} \right) \frac{1+Z_{n,1}(s)}{R^3} \\ -\mu \left(\frac{Y_{n-1,2}}{Y_{n-1,5}} \right) \frac{1+Z_{n,2}(s)}{R^3} \\ -\mu \left(\frac{1}{Y_{n-1,6}} \right) \frac{Z_{n,3}(s)}{R^3} \end{pmatrix}, \tag{7.74}$$

$$G_{n,P}(Z_n) = D_n^{-1} F_P(Y) = D_n^{-1} \begin{pmatrix} 0 \\ 0 \\ 0 \\ A\vec{f_P} \left[A^T \begin{pmatrix} Y_{n-1,1} [1 + Z_{n,1}(s)] \\ Y_{n-1,2} [1 + Z_{n,2}(s)] \\ Z_{n,3}(s) \end{pmatrix} \right] \end{pmatrix}, \tag{7.75}$$

where $\vec{f_P}(.)$ is given in (7.42) and $R = \sqrt{Y_1^2 + Y_2^2 + Y_3^2}$, i.e.:

$$R = \sqrt{Y_{n-1,1}^2 [1 + Z_{n,1}(s)]^2 + Y_{n-1,2}^2 [1 + Z_{n,2}(s)]^2 + [Z_{n,3}(s)]^2}. \tag{7.76}$$

Using the End-Of-Slice condition (7.67) and rescaling with the change of variables (7.70), together with $\beta_n = 1$, make solving the satellite problem (7.68) equivalent to solving, on each n^{th} slice $[0, s_n]$, corresponding to $[T_{n-1}, T_n]$, a rescaled initial value shooting problem in which one seeks for $s_n \in \mathbb{R}$ and $Z_n : [0, s_n] \longrightarrow \mathbb{R}^6$ such that:

$$\begin{cases} \frac{dZ_n}{ds} = G_n(Z_n), \ s > 0 & (7.77.1) \\ Z_n(0) = 0, & (7.77.2) \\ L_z [Z_n(s_n)] = 0 \quad \text{and} \quad \forall s \in (0, s_n), \ L_z [Z_n(s)] \neq 0. & (7.77.3) \end{cases} \tag{7.77}$$

where:

$$G_n(Z_n) = G_{n,K}(Z_n) + G_{n,P}(Z_n), \tag{7.78}$$

$G_{n,K}(Z_n)$ and $G_{n,P}(Z_n)$ being given in (7.74) and (7.75) respectively.

7.4.3 Existence of a Weak Ratio Property

Since the nonzeroness condition (1.11) is not satisfied for the third component of $\{Y_n\}$, we start with the following definition.

Definition 10 ***: Generalized Ratio-Vectors***
The ratio-vector, on the n^{th} slice, is defined by:

$$R_n = D_n^{-1} Y_n. \tag{7.79}$$

In the present case, where $Y_{n,3} = 0, \ (\forall n)$, this general definition is, for all $n > 1$, component wise given as:

$$\begin{cases} R_{n,i} = \frac{Y_{n,i}}{Y_{n-1,i}}, & \text{if } i \neq 3, \\ R_{n,i} = 1, & \text{if } i = 3. \end{cases} \tag{7.80}$$

This means that $R_{n,i}$ is the ratio of the end-of-slice value to the starting value of the solution, for all the nonzero components, and is equal to 1 otherwise. (Note that R_n is not defined for $n = 1$, since $Y_{0,6} = 0$ due to the choice of the xy-plane).
Since $Y_{n,3} = 0$ $(\forall n)$ and $R_{n,3} = 1$, (7.80) yields:

$$\forall n > 1, \ \ \forall i \in \{1, 2, \cdots, 6\}, \ \ Y_{n,i} = R_{n,i} Y_{n-1,i}.$$

This makes the recurrence relation, given in formula (4.6) of chapter 4, keep holding for $n > 1$:

$$Y_n = D_{R_n} Y_{n-1}. \tag{7.81}$$

Rescaled Ratio Vectors:
The change of variable (1.5.2) yields:

$$\text{at } t = T_n, \quad Y_n = Y_{n-1} + D_n Z_n(s_n).$$

This relation becomes:
$Z_n(s_n) = D_n^{-1}(Y_n - Y_{n-1}) = D_n^{-1} Y_n - D_n^{-1} Y_{n-1} = R_n - D_n^{-1} Y_{n-1}$.
This is equivalent to:

$$\forall n > 1, \qquad R_n = D_n^{-1} Y_{n-1} + Z_n(s_n), \tag{7.82}$$

making R_n component wise given, for all $n > 1$, by:

$$\begin{cases} R_{n,i} = 1 + Z_{n,i}(s_n), & \text{if } i \neq 3, \\ R_{n,i} = 1, & \text{if } i = 3. \end{cases} \tag{7.83}$$

Since $Z_{n,3}(s_n) = 0$, one deduces also that formula (4.9), given in Chapter 4, keeps holding too:

$$\forall n, \quad R_n = \mathbf{1} + Z_n(s_n).$$

Then, the recurrence (7.81) rescales to:

$$\forall n, \ Y_n = \left(I + D_{Z_n(s_n)}\right) Y_{n-1}. \tag{7.84}$$

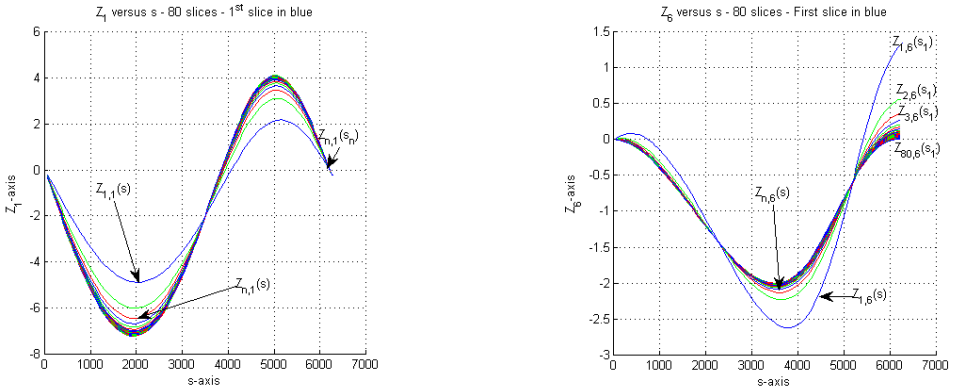
Numerical Ratio Property:
Note first that a Keplerian motion is periodic and would have led, for an EOS condition defining a complete rotation around the earth, to a perfect ratio property.

As stated previously, all perturbing forces are very small compared to the centered force defining the Keplerian motion. It follows that any perturbing force (in particular, that of the J_2-problem) results in a small perturbation of the periodic problem. This yields a sequence of IVP's of which the end-of-slice values $\{Y_n\}$ of the solution present slight variations between two consecutive slices: they start with a value Y_1 that is close to Y_0, then Y_2 that is close to Y_1, and so on...
Translated to the rescaled solutions, those slight variations yield a weak similarity, as described in definition 4 of Chapter 1:

$$\begin{gathered} \exists \epsilon << 1, \ \ \exists n_0 \in \mathbb{N}, \ \ \exists n_r \geq 1, \\ \forall n \in \{n_0 + 1, ..., n_0 + n_r\}, \qquad \|Z_n(s_n) - Z_{n-1}(s_{n-1})\|_\infty < \epsilon. \end{gathered}$$

Figure 7.4.3 gives the graphs of the first and sixth components of Z_n, with respect to s, on 80 consecutive time-slices. The graphs corresponding to each slice alternate the colors: that of the first slice appears clearly in blue. One can notice the slight differences between

the graphs on two consecutive slices and, moreover, how close are the end-of-slice values $Z_n(s_n)$ of the rescaled solution, on all slices.



(a) First component

(b) Sixth component

Figure 7.9: Satellite, J2-perturbed Motion - Z_n versus s on 80 time-slices ($e_0 = 0.1$, $a_0 = 7300$km, $i_0 = 98°$, $\omega_0 = 10°$, $\Omega_0 = 45°$, $M_0 = 123°$).

As stated in proposition 8 of Chapter 4, the weak similarity of the rescaled systems translates to a weak ratio property:

$$\exists \epsilon << 1, \ \ \exists n_0 \in \mathbb{N}, \ \ \exists n_r \geq 1,$$
$$\forall n \in \{n_0 + 1, ..., n_0 + k_s\}, \qquad \|R_n - R_{n-1}\|_\infty < \epsilon.$$

Note that the slight variations of the end-of-slice values $\{Y_n\}$ of the solution between consecutive slices yield:

$$\forall n, \quad \|R_n\|_\infty \approx 1.$$

Note also that for all n, one has $Y_{n,3} = Z_{n,3}(s_n) = 0$ (due to the EOS condition), implying:

$$\forall n, \ \ R_{n,3} = 1.$$

Tables 7.3, 7.4, 7.5 and 7.6 verify numerically the weak ratio property obtained when computing a satellite trajectory, in a J_2-perturbed motion, for 4 different initial conditions. Each initial condition is given by its six Keplerian orbital elements $(a_0, e_0, i_0, \omega_0, \Omega_0, M_0)$. The tables give, for $n = 1, 2, 4, 5$ and 6, the differences $R^i_n - R^i_{n-1}$ of each i^{th} component of consecutive ratios.

One can notice that the ratio property is the weakest for the 6^{th} component.
This is due to the fact that this component, that is zero in a Keplerian motion (due to the choice of the xy-plane of the IPQW-frame), expresses strongly the perturbation of the motion, especially on the first slices.
However, the numerical experiments show, not surprisingly, that as soon as Y_3 reaches the same order of magnitude than Y_1 and Y_2, and Y_6 reaches the same order of magnitude than Y_4 and Y_5, the ratio property becomes also of the same order for all components.

Case A					
$e_0 = 0.1, \quad a_0 = 7300$ km, $\quad i_0 = 98°, \quad \omega_0 = 10°, \quad \Omega_0 = 45°, \quad M_0 = 123°$					
n	$R_n^1 - R_{n-1}^1$	$R_n^2 - R_{n-1}^2$	$R_n^4 - R_{n-1}^4$	$R_n^5 - R_{n-1}^5$	$R_n^6 - R_{n-1}^6$
2	1.44618E-01	-1.97842E-02	-1.48477E-02	1.14859E-01	-7.80803E-01
3	3.97542E-02	-3.39519E-03	-2.41280E-03	2.89542E-02	-2.05384E-01
4	1.64725E-02	-1.18967E-03	-8.26582E-04	1.15840E-02	-9.56773E-02
⋮	⋮	⋮	⋮	⋮	⋮
51	8.39849E-06	-3.08709E-07	-1.43054E-06	-4.48296E-07	-4.12030E-04
52	7.90664E-06	-2.89081E-07	-1.40890E-06	-7.61838E-07	-3.95821E-04
53	7.45149E-06	-2.71338E-07	-1.38848E-06	-1.05147E-06	-3.80550E-04
54	7.02969E-06	-2.55305E-07	-1.36917E-06	-1.31939E-06	-3.66147E-04
55	6.63815E-06	-2.40825E-07	-1.35086E-06	-1.56760E-06	-3.52548E-04
⋮	⋮	⋮	⋮	⋮	⋮
196	-6.33299E-07	-6.65711E-07	-3.60991E-07	-5.60294E-06	-2.62486E-05
197	-6.40445E-07	-6.70923E-07	-3.55889E-07	-5.60425E-06	-2.59728E-05
198	-6.47544E-07	-6.76132E-07	-3.50801E-07	-5.60555E-06	-2.57012E-05
199	-6.54579E-07	-6.81338E-07	-3.45728E-07	-5.60682E-06	-2.54336E-05
200	-6.61575E-07	-6.86540E-07	-3.40670E-07	-5.60809E-06	-2.51699E-05

Table 7.3: Satellite, J2-perturbed Motion - Ratio Property, (A)

Case B					
$e_0 = 0.1, \quad a_0 = 7650$ km, $\quad i_0 = 98°, \quad \omega_0 = 10°, \quad \Omega_0 = 45°, \quad M_0 = 123°$					
n	$R_n^1 - R_{n-1}^1$	$R_n^2 - R_{n-1}^2$	$R_n^4 - R_{n-1}^4$	$R_n^5 - R_{n-1}^5$	$R_n^6 - R_{n-1}^6$
2	1.44600E-01	-1.97855E-02	-1.48502E-02	1.14848E-01	-7.80802E-01
3	3.97494E-02	-3.39555E-03	-2.41297E-03	2.89522E-02	-2.05388E-01
4	1.64705E-02	-1.18984E-03	-8.26476E-04	1.15838E-02	-9.56793E-02
⋮	⋮	⋮	⋮	⋮	⋮
51	8.42450E-06	-3.09811E-07	-1.25374E-06	5.50658E-07	-4.11970E-04
52	7.93387E-06	-2.88918E-07	-1.23347E-06	2.36388E-07	-3.95762E-04
53	7.47994E-06	-2.69906E-07	-1.21442E-06	-5.39747E-08	-3.80492E-04
54	7.05935E-06	-2.52604E-07	-1.19648E-06	-3.22618E-07	-3.66090E-04
55	6.66903E-06	-2.36852E-07	-1.17955E-06	-5.71543E-07	-3.52491E-04
⋮	⋮	⋮	⋮	⋮	⋮
196	-4.28889E-07	-4.76154E-07	-3.73518E-07	-4.63678E-06	-2.63654E-05
197	-4.34866E-07	-4.80122E-07	-3.69491E-07	-4.63786E-06	-2.60909E-05
198	-4.40805E-07	-4.84090E-07	-3.65473E-07	-4.63894E-06	-2.58205E-05
199	-4.46684E-07	-4.88058E-07	-3.61464E-07	-4.63999E-06	-2.55541E-05
200	-4.52527E-07	-4.92025E-07	-3.57463E-07	-4.64103E-06	-2.52917E-05

Table 7.4: Satellite, J2-perturbed Motion - Ratio Property, (B)

Case C					
$e_0 = 0.0005, \quad a_0 = 7300$ km, $\quad i_0 = 98°, \quad \omega_0 = 10°, \quad \Omega_0 = 45°, \quad M_0 = 123°$					
n	$R_n^1 - R_{n-1}^1$	$R_n^2 - R_{n-1}^2$	$R_n^4 - R_{n-1}^4$	$R_n^5 - R_{n-1}^5$	$R_n^6 - R_{n-1}^6$
2	1.25554E-01	-9.98703E-03	-9.98222E-03	1.25577E-01	-7.49305E-01
3	3.33172E-02	-1.73365E-03	-1.73439E-03	3.33241E-02	-2.02539E-01
4	1.35964E-02	-6.07054E-04	-6.08181E-04	1.35988E-02	-9.49112E-02
⋮	⋮	⋮	⋮	⋮	⋮
51	6.56125E-06	-2.16403E-07	-1.41967E-06	5.36804E-06	-4.11892E-04
52	6.17347E-06	-2.03936E-07	-1.40735E-06	4.97991E-06	-3.95692E-04
53	5.81498E-06	-1.92416E-07	-1.39599E-06	4.62106E-06	-3.80429E-04
54	5.48306E-06	-1.81757E-07	-1.38550E-06	4.28881E-06	-3.66034E-04
55	5.17530E-06	-1.71878E-07	-1.37578E-06	3.98071E-06	-3.52441E-04
⋮	⋮	⋮	⋮	⋮	⋮
196	-3.67265E-09	-6.29744E-09	-1.26205E-06	-1.26196E-06	-2.68958E-05
197	-5.07409E-09	-6.25458E-09	-1.26258E-06	-1.26401E-06	-2.66253E-05
198	-6.43648E-09	-6.21307E-09	-1.26313E-06	-1.26603E-06	-2.63589E-05
199	-7.76905E-09	-6.17259E-09	-1.26367E-06	-1.26802E-06	-2.60965E-05
200	-9.07144E-09	-6.13313E-09	-1.26422E-06	-1.26998E-06	-2.58381E-05

Table 7.5: Satellite, J2-perturbed Motion - Ratio Property, (C)

Case D					
$e_0 = 0.1, \quad a_0 = 7300$ km, $\quad i_0 = 110°, \quad \omega_0 = 10°, \quad \Omega_0 = 45°, \quad M_0 = 123°$					
n	$R_n^1 - R_{n-1}^1$	$R_n^2 - R_{n-1}^2$	$R_n^4 - R_{n-1}^4$	$R_n^5 - R_{n-1}^5$	$R_n^6 - R_{n-1}^6$
2	1.44961E-01	-1.97825E-02	-1.48547E-02	1.15001E-01	-7.80836E-01
3	3.98526E-02	-3.39565E-03	-2.41889E-03	2.89880E-02	-2.05407E-01
4	1.65106E-02	-1.19016E-03	-8.32490E-04	1.15954E-02	-9.56906E-02
⋮	⋮	⋮	⋮	⋮	⋮
51	3.11166E-06	-5.92193E-07	-7.60738E-06	-3.70349E-06	-4.13966E-04
52	2.65506E-06	-5.68757E-07	-7.59624E-06	-4.01049E-06	-3.97759E-04
53	2.23502E-06	-5.47203E-07	-7.58642E-06	-4.29368E-06	-3.82491E-04
54	1.84818E-06	-5.27357E-07	-7.57781E-06	-4.55525E-06	-3.68090E-04
55	1.49147E-06	-5.09061E-07	-7.57031E-06	-4.79720E-06	-3.54493E-04
⋮	⋮	⋮	⋮	⋮	⋮
196	-2.29309E-06	-4.11479E-07	-9.34435E-06	-9.15482E-06	-2.88588E-05
197	-2.28286E-06	-4.13464E-07	-9.37028E-06	-9.16850E-06	-2.85883E-05
198	-2.27266E-06	-4.15458E-07	-9.39644E-06	-9.18234E-06	-2.83219E-05
199	-2.26248E-06	-4.17462E-07	-9.42284E-06	-9.19636E-06	-2.80596E-05
200	-2.25232E-06	-4.19475E-07	-9.44948E-06	-9.21056E-06	-2.78012E-05

Table 7.6: Satellite, J2-perturbed Motion - Ratio Property, (D)

7.4.4 Numerical Results: Application of RaPTI

Tables (7.7), (7.8), (7.9) and (7.10) below summarize the numerical results that were obtained by applying RaPTI algorithm for computing a satellite trajectory, in a J_2-perturbed

motion, for 12 different initial conditions. Each initial condition is given by its six Keplerian orbital elements $(a_0, e_0, i_0, \omega_0, \Omega_0, M_0)$.

The mathematical model that has been used for the predictions of the ratios, resulting from implementing the procedure FIND_RATIOS_MODEL in the preliminary analysis, is a polynomial of degree $d = 2$ obtained by fitting the last $n_{prec} = 3$ computed ratios, in the least-squares sense.

Knowing that the sizes of successive time-slices of this quasi-periodic problem differ very slightly from the common size ΔT_{per} of all slices of the corresponding periodic problem, the total number of slices N has been approximated by:

$$N = \left\lceil \frac{T}{\Delta T_{per}} \right\rceil,$$

where T is the desired time of integration.
In average on the 12 cases, $\Delta T_{per} \approx 0.72$ days. The experiments have been done for $N = 600$ (corresponding to about 43 days) and for $N = 1500$ (corresponding to about 108 days).

The total number of iterations vary from one case to another, but here also, the results show that this number is small, compared to the total number of slices, in spite of the absence of any strong similarity property.

In all the experiments:
- The time-step of integration that has been used is $\tau = 60$ seconds.
- The tolerance up to which is reached the EOS condition is $\epsilon_{tol}^{eos} = 10^{-9}$.
- The tolerance used for getting n_s is $\epsilon_{tol}^{n_s} = 5 \times 10^{-4}$.
- The minimum value for n_s is $n_{s,min} = 20$.
- The tolerance on the relative gaps is $\epsilon_{tol}^{g} = 10^{-5}$.

Case	1	2	3	4	5	6
e_0	0.1	0.1	0.1	0.0005	0.15	0.1
a_0	7300 km	7650 km	8000 km	7300 km	7300 km	7300 km
i_0	98°	98°	98°	98°	98°	80°
ω_0	10°	10°	10°	10°	10°	10°
Ω_0	45°	45°	45°	45°	45°	45°
M_0	123°	123°	123°	123°	123°	123°
N	**600**	**600**	**600**	**600**	**600**	**600**
n_{days}	43.12	46.26	49.47	43.15	43.11	43.12
$\mathbf{T_s}$	**26.2**	**26.84**	**28.03**	**25.61**	**24.41**	**25.83**
n_s	46	46	46	46	46	46
n_I	10	9	9	10	10	11
T_2	16.94	17.44	17.97	16.79	17	17.06
$\mathbf{E_2}$	**0.77**	**0.77**	**0.78**	**0.76**	**0.72**	**0.76**
$\mathbf{S_2}$	**1.55**	**1.54**	**1.56**	**1.53**	**1.44**	**1.51**
S_2^{max}	1.86	1.86	1.86	1.86	1.86	1.86
T_4	11.56	11.89	12.52	11.63	11.66	11.97
$\mathbf{E_4}$	**0.57**	**0.56**	**0.56**	**0.55**	**0.52**	**0.54**
$\mathbf{S_4}$	**2.27**	**2.26**	**2.24**	**2.20**	**2.09**	**2.16**
S_4^{max}	3.25	3.25	3.25	3.25	3.25	3.25
T_8	10.31	10.31	10.84	10.23	10.42	10.82
$\mathbf{E_8}$	**0.32**	**0.33**	**0.32**	**0.31**	**0.29**	**0.30**
$\mathbf{S_8}$	**2.54**	**2.60**	**2.59**	**2.50**	**2.34**	**2.39**
S_8^{max}	5.21	5.21	5.21	5.21	5.21	5.21

Table 7.7: Satellite, J2-perturbed Motion - Application of RaPTI, $N = 600$ (1).

Case	7	8	9	10	11	12
e_0	0.1	0.1	0.1	0.1	0.1	0.1
a_0	7300 km	7300 km	7300 km	7300 km	7300 km	7300 km
i_0	98°	98°	98°	98°	98°	98°
ω_0	5°	20°	10°	10°	10°	10°
Ω_0	45°	45°	10°	120°	45°	45°
M_0	123°	123°	123°	123°	20°	60°
N	**600**	**600**	**600**	**600**	**600**	**600**
n_{days}	43.14	43.1	43.13	43.13	43.14	43.25
$\mathbf{T_s}$	**25.71**	**25.94**	**25.54**	**25.95**	**25.34**	**25.86**
n_s	46	46	46	46	46	46
n_I	10	10	10	10	11	10
T_2	16.71	16.95	16.68	16.82	16.94	17.05
$\mathbf{E_2}$	**0.77**	**0.77**	**0.77**	**0.77**	**0.75**	**0.76**
$\mathbf{S_2}$	**1.54**	**1.53**	**1.53**	**1.54**	**1.50**	**1.52**
S_2^{max}	1.86	1.86	1.86	1.86	1.86	1.86
T_4	11.61	11.75	11.49	11.57	11.85	11.53
$\mathbf{E_4}$	**0.55**	**0.55**	**0.56**	**0.56**	**0.53**	**0.56**
$\mathbf{S_4}$	**2.21**	**2.21**	**2.22**	**2.24**	**2.14**	**2.24**
S_4^{max}	3.25	3.25	3.25	3.25	3.25	3.25
T_8	10.37	10.41	10.31	10.37	10.8	10.43
$\mathbf{E_8}$	**0.31**	**0.31**	**0.31**	**0.31**	**0.29**	**0.31**
$\mathbf{S_8}$	**2.48**	**2.49**	**2.48**	**2.50**	**2.35**	**2.48**
S_8^{max}	5.21	5.21	5.21	5.21	5.21	5.21

Table 7.8: Satellite, J2-perturbed Motion - Application of RaPTI, $N = 600$ (2).

RaPTI has been tested on the previous 12 cases, on 600 slices, using 2, 3, 4, 5, 6, 7, and 8 processors. Figure 7.10 shows how the values of the efficiency and speed-up, averaged on the 12 cases, vary with the number of processors.

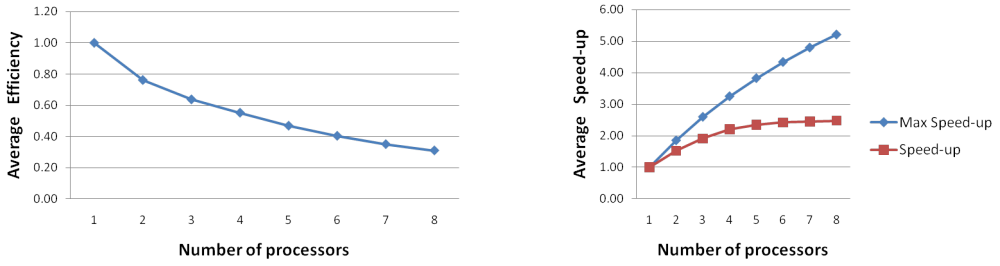


Figure 7.10: Satellite, J2-perturbed Motion - Efficiencies and Speed-ups, Averaged on the 12 Cases, for $N = 600$.

Case	1	2	3	4	5	6
e_0	0.1	0.1	0.1	0.0005	0.15	0.1
a_0	7300 km	7650 km	8000 km	7300 km	7300 km	7300 km
i_0	98°	98°	98°	98°	98°	80°
ω_0	10°	10°	10°	10°	10°	10°
Ω_0	45°	45°	45°	45°	45°	45°
M_0	123°	123°	123°	123°	123°	123°
N	**1500**	**1500**	**1500**	**1500**	**1500**	**1500**
n_{days}	107.81	115.65	123.68	107.88	107.78	107.81
n_s	46	46	46	46	46	46
n_I	18	15	14	17	22	48
$\mathbf{T_s}$	**64.17**	**67.70**	**70.96**	**64.35**	**64.02**	**64.41**
T_2	40.48	41.95	44.17	39.67	41.06	43.35
$\mathbf{E_2}$	**0.79**	**0.81**	**0.80**	**0.81**	**0.78**	**0.74**
$\mathbf{S_2}$	**1.59**	**1.61**	**1.61**	**1.62**	**1.56**	**1.49**
S_2^{max}	1.94	1.94	1.94	1.94	1.94	1.94
T_4	24.95	25.31	26.36	24.51	26.03	30.20
$\mathbf{E_4}$	**0.64**	**0.67**	**0.67**	**0.66**	**0.61**	**0.53**
$\mathbf{S_4}$	**2.57**	**2.67**	**2.69**	**2.63**	**2.46**	**2.13**
S_4^{max}	3.66	3.66	3.66	3.66	3.66	3.66
T_8	19.96	19.45	20.02	19.35	22.29	28.95
$\mathbf{E_8}$	**0.40**	**0.44**	**0.44**	**0.42**	**0.36**	**0.28**
$\mathbf{S_8}$	**3.21**	**3.48**	**3.54**	**3.33**	**2.87**	**2.22**
S_8^{max}	6.59	6.59	6.59	6.59	6.59	6.59

Table 7.9: Satellite, J2-perturbed Motion - Application of RaPTI, $N = 1500$ (1).

Case	7	8	9	10	11	12
e_0	0.1	0.1	0.1	0.1	0.1	0.1
a_0	7300 km	7300 km	7300 km	7300 km	7300 km	7300 km
i_0	98°	98°	98°	98°	98°	98°
ω_0	5°	20°	10°	10°	10°	10°
Ω_0	45°	45°	10°	120°	45°	45°
M_0	123°	123°	123°	123°	20°	60°
N	**1500**	**1500**	**1500**	**1500**	**1500**	**1500**
n_{days}	108.08	108.11	107.81	107.81	107.81	108.10
$\mathbf{T_s}$	**65.16**	**65.06**	**63.65**	**64.85**	**64.81**	**64.16**
n_s	46	46	46	46	46	46
n_I	34	25	18	18	17	18
T_2	42.08	41.00	40.46	40.45	40.38	40.22
$\mathbf{E_2}$	**0.77**	**0.79**	**0.79**	**0.80**	**0.80**	**0.80**
$\mathbf{S_2}$	**1.55**	**1.59**	**1.57**	**1.60**	**1.60**	**1.60**
S_2^{max}	1.94	1.94	1.94	1.94	1.94	1.94
T_4	27.25	25.84	24.93	24.78	24.60	24.75
$\mathbf{E_4}$	**0.60**	**0.63**	**0.64**	**0.65**	**0.66**	**0.65**
$\mathbf{S_4}$	**2.39**	**2.52**	**2.55**	**2.62**	**2.63**	**2.59**
S_4^{max}	3.66	3.66	3.66	3.66	3.66	3.66
T_8	23.68	20.37	19.96	19.87	19.33	20.21
$\mathbf{E_8}$	**0.34**	**0.40**	**0.40**	**0.41**	**0.42**	**0.40**
$\mathbf{S_8}$	**2.75**	**3.19**	**3.19**	**3.26**	**3.35**	**3.17**
S_8^{max}	6.59	6.59	6.59	6.59	6.59	6.59

Table 7.10: Satellite, J2-perturbed Motion - Application of RaPTI, $N = 1500$ (2).

RaPTI has been tested on the previous 12 cases, on 1500 slices, using 2, 3, 4, 5, 6, 7, and 8 processors. Figure 7.11 shows how the values of the efficiency and speed-up, averaged on the 12 cases, vary with the number of processors.

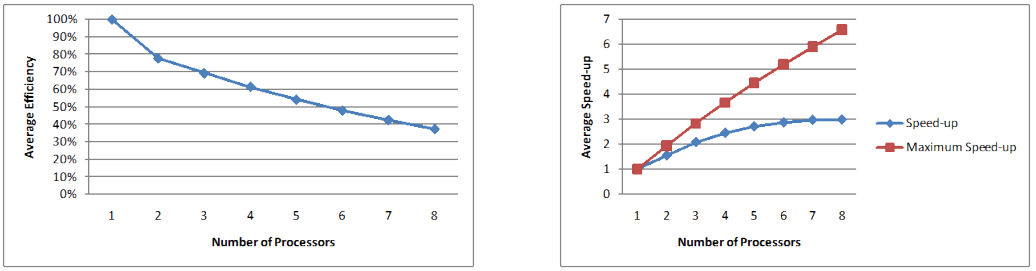


Figure 7.11: Satellite, J2-perturbed Motion - Efficiencies and Speed-ups, Averaged on the 12 Cases, for $N = 1500$.

One can notice that the absence of asymptotic ratio property makes the application of RaPTI yield a speed-up that is quite far from the maximum speed-up stated by Amdahl's law.
However, since the goal of time-parallelism is to tackle real-time problems, any speed-up is welcome.

Chapter 8

Conclusion & Perspectives

From Rescaling to RaPTI Algorithm

In this thesis, we have devised a new approach for solving, in a time-parallel way, the first order Initial Value Problems: $\frac{dY}{dt} = F(Y),\ t > 0,\ \ Y(0) = Y_0$, assumed to have a unique and continuous solution.
A rescaling methodology underlies our approach. It is based on (i) an end-of-slice condition that generates automatically a coarse grid of time slices and (ii) a rescaling technique that changes both the time variable t and the solution $Y(t)$ into a rescaled time variable s and a rescaled solution $Z_n(s)$, setting them both to zero at the onset of every slice. The result is a sequence of rescaled problems that can be solved by means of a standard numerical method.
For some choices of the end-of-slice condition and time-rescaling factors, one obtains a uniform similarity of the rescaled problems that adds to the method a robustness that makes it well-suited to tackle stiff problems.
Some other choices might yield an invariance, or near-invariance, of the rescaled problems that result in a ratio property. This provides a ratio-based prediction procedure of the starting values of the solution.
Since rescaling sets the time at 0 at the onset of every slice, parallel computations can then be done for solving the rescaled problems on different slices, within the local time and starting with the predicted initial values.
The result is a Ratio-based Parallel Time Integration (RaPTI) algorithm.

RaPTI algorithm has been successfully applied to problems on which the rescaling methodology has led to the following cases of similarity:
(i) Invariance (membrane problem, Keplerian satellite problem): It has shown a perfect parallelism with no iterations and no communication; perfect to such a point that no parallelism is needed!
(ii) Asymptotic similarity (linear problems, membrane problem, reaction-diffusion problem): the algorithm converges in a very small number of iterations; the speed-up is very good and quite close to the maximum speed-up stated by Amdahl's law.
(iii) Weak similarity (J2-perturbed satellite problem): the algorithm converges in a relatively small number of iterations; the speed-up is quite acceptable.

Scope of Application of RaPTI Algorithm

As shown throughout this thesis, the application of RaPTI Algorithm for solving initial value problems in a time-parallel way, is not unconditional and requires the following conditions to be satisfied:

1. *Existence of an EOS condition defining a coarse grid:*
   The choice of an appropriate EOS condition is problem-dependent and necessitates the *prior knowledge of the behavior of the solution.* Moreover, this behavior should be likely to induce an EOS condition yielding, in a unique way, an increasing sequence $\{T_n\}_{n\geq 0}$ satisfying $\cup_{n\geq 1}[T_{n-1}, T_n] = [0, T]_{T\leq\infty}$.

2. *Existence of a ratio-property:*
   The existence of a ratio-property is essential for yielding good predictions and therefore fast convergence of RaPTI algorithm. This is possible if one can combine the choice of a relevant EOS condition with that of time-rescaling factors, in a way that yields invariance or near-invariance properties.

Those two constraints determine and reduce the scope of application of RaPTI Algorithm.

Relevance of RaPTI Algorithm

When RaPTI algorithm is applicable, and as shown in the numerical results given in chapters 5, 6 and 7, the number of iterations is *much less* than the number of slices and the resulting speed-up is close to the maximum speed-up stated by Amdahl's law. This is mainly due to the following facts:

1. *Accuracy of the ratio-based predictions:*
   By following the path of the solution, a relevant EOS condition generates a coarse grid that is well adapted to the behavior of the solution, and yields a sequence $\{Y_n\}$ satisfying a ratio property.
   The ratio-based prediction procedure provides then:
   - exact predictions, in case of invariance,
   - very accurate predictions, in case of asymptotic similarity,
   - quite good predictions, in case of weak similarity.

2. *Predictions do not require any sequential integration on the coarse grid:*
   The only operations involved in the ratio-based prediction procedure are not costly and consist of a polynomial fit of a few ratio-vectors that is used for the evaluation (by extrapolation) of the remaining ratios. Note also that the corrective step that updates the predictions, at each iteration, consists simply of new predictions based on the last converged ratio-vectors. However, RaPTI algorithm needs the preliminary sequential integration on a small number n_s of slices on which will rely the ratio-based predictions of the first iteration.

3. *Not all the time-slices are solved at each iteration:*
   In general, time-parallel schemes, such as multiple shooting methods or parareal algorithm, need to solve in parallel and at each iteration all the not-yet-converged time-slices of the coarse grid, in order to evaluate the jumps and apply the corrective step that is intended to update the predictions. Whereas RaPTI algorithm solves the time-slices as long as the resulting jumps are within the desired tolerance. It *stops the*

parallel computations as soon as a slice fails the test of convergence thus, reducing the total overhead coming from parallel computations. This is due to the fact that the ratio-based prediction procedure requires nothing but the last few convergent ratios.

Hence, when it applies, RaPTI algorithm is likely to outperform other time-parallel schemes.

Perspectives

A good speed-up is the main goal of any parallel-in-time algorithm.
Because the ratio-based prediction procedure of RaPTI algorithm is likely to provide quite good predictions and to require very little sequential computations, and since good predictions and little sequential computations are the main key for a time-parallel scheme to yield a fast convergence and a high speed-up, RaPTI algorithm seems to have a great potential in time-parallelism.
Future work on the subject could have the following objectives:

1. *Mathematical Investigation of the convergence of the method:*
   This convergence is not completely proved. The difficulty lies in the fact that the predictions are done on a coarse grid that is unknown! Therefore, both sequences $\{Y_n^c\}$ and $\{T_n^c\}$, of the corrected end-of-slice values of the solution and the time, should be proved to converge to their exact values. However, since the convergence is numerically verified (by comparing the values obtained with RaPTI algorithm to those obtained with a sequential computation), further investigations are likely to confirm this convergence with a mathematical proof, under some relevant conditions. An *error analysis* should follow and enhance the relevance of RaPTI.

2. *Experimentation of problems where the solution has an attraction point at ∞:*
   A relevant EOS condition has been proposed in Chapter 2. Interesting application problems should be found and experimented.

3. *Determination of the exact scope of application of RaPTI:*
   All possible behaviors should be considered, in order to check, for each case, if one can find EOS conditions generating a coarse grid and a sequence $\{Y_n\}$ presenting a ratio property.

4. *Comparative analysis:*
   Our research will not be convincing before we apply RaPTI algorithm to problems on which other time-parallel algorithms have been implemented, in order to compare the resulting speed-ups.

The first of these objectives seems to have priority and will be probably be our next target.

Chapter 9

Appendix 1: Procedures used in RaPTI

1. **Procedure DETECT_RATIO_PROPERTY_UP_TO_ϵ:**

   For given rescaled systems (S'_n) and a given tolerance ϵ, this procedure keeps solving sequentially and stops when (i) a ratio-property is detected at slice number n_s or (ii) a maximum number n_{max} of slices is solved with no ratio property numerically detected.

```
Initialize n = 1
SOLVE_SLICE                          % Procedure described in item 11
Initialize Test_R = 2ε
Label 1     While Test_R > ε and n ≤ n_stop − n_r
                n=n+1
                SOLVE_SLICE
                Test_R = ‖R_n − R_{n−1}‖_∞
            End While
            n_0 = n.
            If Test_R ≤ ε
                While Test_R ≤ ε and n ≤ n_0 + n_r
                    n=n+1
                    SOLVE_SLICE
                    Test_R = ‖R_n − R_{n−1}‖_∞
                End While
                If n = n_0 + n_r
                   n_s = n          % Ratio property numerically detected
                Else if n < n_stop − n_r
                   Goto Label 1 % Ratio property not holding on n_r slices
                Else
                   "No ratio property"
                End if
            Else
                "No ratio property"
            End if
```

2. **Procedure REACH_ASYMPTOTIC_RATIO_PROPERTY_UP_TO_ϵ:**

For given rescaled systems (S'_n), a given limit problem S_L and a given tolerance ϵ, this procedure solves the limit problem and keeps solving (S'_n) sequentially until the asymptotic ratio-property is reached at slice number n_s.

SOLVE_SLICE_LIMIT *% Procedure described in item 10*
Initialize $n = 1$
SOLVE_SLICE *% Procedure described in item 11*
Initialize $Test\_R = 2\epsilon$
Label 1 *While* $Test\_R > \epsilon$
n=n+1
SOLVE_SLICE
$Test\_R = \|R_n - R_L\|_\infty$
End While
$n_0 = n$.
While $Test\_R \leq \epsilon$ and $n \leq n_0 + n_r$
n=n+1
SOLVE_SLICE
$Test\_R = \|R_n - R_L\|_\infty$
End While
If $n = n_0 + n_r$
$n_s = n$ *% Ratio property reached*
Else
Goto Label 1 *% Ratio property not holding on n_r slices*
End if

3. **Procedure FIND_RATIOS_MODEL:**

Given n_M models R_{fit_j} $(1 \leq j \leq n_M)$ fitting the exact ratios $\{R_n\}_{1<n\leq n_s}$, and given a validation set $\{R_{n_s+1}, \cdots, R_{n_s+n_{add}}\}$ of n_{add} additional exact ratio-vectors, this procedure choses the model extrapolating best on the n_{add} slices, thus providing the predictive mathematical model R_{fit} to be used.

Initialize $test\_error = \infty$
For $j = 1$ *to* n_M
For $n = n_s + 1$ *to* $n_s + n_{add}$
Evaluate $R_{fit_j}(n)$,
End For
$error_j = \max_{n_s<n\leq n_s+n_{add}} \left[\max_{i\in\{1,2,\cdots,K\}} \left|R_{fit_j,i}(n) - R_{n,i}\right|\right]$.
If $error_j < test\_error$ *% Extrapolation Test*
$R_{fit} = R_{fit_j}$
$test\_error = error_j$
End If
End For

4. **Procedure PREDICT_AND_ESTIMATE_N:**

When a refined estimation of the total number of slices N is required, this procedure is duplicated on all processors, at the beginning of each k^{th} iteration.
Given the number $n_s^{(k-1)}$ of slices having converged before this iteration and given the interval $\left[0, T_{n_s^{(k-1)}}\right]$ on which the problem has been so far solved, one gets the new predictions $\{Y_n^{p^{(k)}}\}_{n>n_s^{(k-1)}}$ together with the estimated cumulative time $\tilde{T}$, until reaching T and without overstepping Y_{max}.
The last value of n gives then an estimation of the total number of slices N.

Initialize $\tilde{T} = T_{n_s^{(k-1)}}$, $n = n_s^{(k-1)}$ and $Y^{p^{(k)}}_{n_s^{(k-1)}} = Y_{n_s^{(k-1)}}$
While $\tilde{T} < T$
    $n = n + 1$
    Predict $R_n^{p^{(k)}} = R_{fit}^{(k)}(n)$ and $Y_n^{p^{(k)}} = D_{R_n^{p^{(k)}}} Y_{n-1}^{p^{(k)}}$
    if $Y_n^{p^{(k)}} < Y_{max}$
        predict $\beta_n^{p^{(k)}}$ in terms of $Y_{n-1}^{p^{(k)}}$
        $\tilde{T} = \tilde{T} + \beta_n^{p^{(k)}} \tilde{s_n}$
    else
        $n = n - 1$ % *The solution oversteps* Y_{max} *before reaching* T
        break
    end if
end while
$N = n$.

5. **Procedure "GET_MODEL_PARAMETERS":**

Given a chosen model R_{fit}, this procedure is duplicated on all processors, at the beginning of each k^{th} iteration.
It reevaluates the model's parameters that fit best the last n_{prec} ratio-vectors, in the least-squares sense (function "*Update_Model*").
The resulting mathematical model, at the k^{th} iteration, is denoted by $R_{fit}^{(k)}$.

For $i = 1$ *to* n_{prec}
    $X(i) = R_{n_s^{(k-1)} - n_{prec} + i}$
End For
$R_{fit}^{(k)} = Update\_Model\,[X(1), \cdots, X(n_{prec})]$

6. **Procedure "PREDICT":**

This procedure is duplicated on all processors, at the beginning of each k^{th} iteration. It uses the mathematical model $R_{fit}^{(k)}$ for getting predicted ratios $\{R_n^p\}_{n>n_s^{(k-1)}}$ and deduces the predicted starting values $\{Y_n\}_{n>n_s^{(k-1)}}$ of the solution.

Initialize $Y^{p^{(k)}}_{n_s^{(k-1)}} = Y^c_{n_s^{(k-1)}}$.
For $n = n_s^{(k-1)} + 1$ *to* N
    $R_n^{p^{(k)}} = R_{fit}^{(k)}(n)$,
    $Y_n^{p^{(k)}} = D_{R_n^{p^{(k)}}} Y_{n-1}^{p^{(k)}}$,
End For

7. **Procedure SOLVE_MY_SLICES_IN_PARALLEL:**

At each k^{th} iteration, this procedure is implemented on all processors.
For $n > n_s^{(k-1)}$, each processor j solves the time slices that are assigned to it (according to the cyclic distribution) and stops solving as soon as the convergence test stops holding, i.e. at slice number $n_{div}^k(j)$.

Initialize $n = n_{first}^{(k)}(j)$ *% First slice assigned to j, at the* k^{th} *iteration*
Initialize $n_{div}^{(k)}(j) = n_{last}^{(k)}(j)$ *% Last slice assigned to j*
While $n \leq N$
    SOLVE_SLICE_PARALLEL *% Procedure described in item 12*
    TEST_SLICE_CONVERGENCE *% Procedure described in item 8*
    If TEST_SLICE_CONVERGENCE = True
        $n = n + n_p$ *% Converging slice... continue solving!*
    Else
        $n_{div}^{(k)}(j) = n$ *% A slice diverges... stop solving.*
        Break
    End If
End While

One can devise a variant of this procedure in a way that, as soon as a processor has a divergent slice, it sends a flag to all other processors, asking to end the iteration. This would necessitate to check for a flag before solving every slice, inducing more communications, and has been said to be useless because of the cyclic distribution.

8. **Procedure TEST_SLICE_CONVERGENCE:**

When the time slices are solved in parallel, using predicted starting values, this procedure evaluates the relative gap (GAP_n) between the predicted and corrected values of Y_n, at the end of each n^{th} solved slice.
If $\|GAP_n\|_\infty \leq \epsilon^g_{tol}$ (ϵ^g_{tol} being a given tolerance on relative gaps), the value "true" is assigned to a logical variable.
Else, the value "false" is assigned to the variable.

$GAP_n = \frac{Y^c_n - Y^{p^{(k)}}_n}{\max\left(\|Y^c_n\|_\infty, \left\|Y^{p^{(k)}}_n\right\|_\infty\right)}$
If $\|GAP_n\|_\infty > \epsilon^g_{tol}$
    TEST_SLICE_CONVERGENCE = False,
Else
    TEST_SLICE_CONVERGENCE = True,
End

9. **Procedure DETERMINE_$n_s^{(k)}$_AND_$\{T^c_n\}$:**

This procedure is implemented on the master processor only (processor 1).
After having received, at the end of each k^{th} iteration and from each processor j ($2 \leq j \leq n_p$), the number $n^{(k)}_{div}(j)$ of its divergent slice and the sizes $\{\Delta T^c_n = \beta^p_n s^c_n\}$ of its solved slices, the master processor evaluates:
(i) the number $n^{(k)}_s$ of the last slice having globally converged (on all processors),
(ii) the starting times $\{T^c_n\}$, for $n^{(k-1)}_s + 1 \leq n \leq n^{(k)}_s$.

divergence $= n^{(k)}_{div}(1)$
For $j = 2$ *to* n_p
    If $n^{(k)}_{div}(j) <$ *divergence*
        divergence $= n^{(k)}_{div}(j)$
    End If
End For
$n^{(k)}_s =$ *divergence*
For $n = n^{(k-1)}_s + 1$ *to* $n^{(k)}_s$
    $T^c_n = T^c_{n-1} + \Delta T^c_n$
End For

10. **Procedure SOLVE_SLICE_LIMIT:**

This procedure is exactly similar to SOLVE_SLICE (see below), but solves the limit problem on one slice (i.e. it uses the function G_L instead of G_n).

11. **Procedure SOLVE_SLICE:**
Given a chosen numerical method, a rescaled time-step τ, a function G_n, an initial value Y_n of the solution, and an initial time T_n, this procedure is duplicated on all processors, in the initialization stage, for solving each of the first n_s slices.

Evaluate β_n and D_n
Initialize $j = 0$, $s_0 = 0$ and $Z_0 = 0$
While $\tau > \epsilon_{tol}^{eos}$ and EOS is not overstepped
    $temp\_Z = Num(G_n, Z_j, \tau)$
    if EOS is not overstepped
        $Z_{j+1} = temp\_Z$
        $s_{j+1} = s_j + \tau$
        $Y_{j+1} = D_n\left(\mathbf{1} + Z_{j+1}\right)$ % *Solution on fine grid*
        $t_{j+1} = T_{n-1} + \beta_n s_{j+1}$ % *Time variable on fine grid*
    else
        $\tau = \frac{\tau}{2}$
    End If
End While
$Y_n = Y_{j+1}$ % *End-of-slice value of the solution*
$T_n = t_{j+1}$ % *End-of-slice value of the time variable*

12. **Procedure SOLVE_SLICE_PARALLEL:**
This procedure is similar to SOLVE_SLICE, but solves a slice, using predicted starting values, i.e. using the functions $\{G_n^p\}$ instead of $\{G_n\}$.
Besides, each processor stocks the solution of the time slices that are assigned to it and *within the local time* s, only. It evaluates however the size $\Delta T_n^c = \beta_n^p s_n^c = T_n^c - T_{n-1}^c$ of each of its slices, with respect to the real time t.

Evaluate β_n^p and D_n^p
Initialize $j = 0$, $s_0^c = 0$ and $Z_0^c = 0$.
While $\tau > \epsilon_{tol}^{eos}$ and EOS is not overstepped
    $temp\_Z = Num(G_n^p, Z_j^c, \tau)$
    if EOS is not overstepped
        $Z_{j+1}^c = temp\_Z$
        $s_{j+1}^c = s_j + \tau$
        $Y_{j+1}^c = D_n^p\left(\mathbf{1} + Z_{j+1}^c\right)$ % *Solution on fine grid*
    else
        $\tau = \frac{\tau}{2}$
    End If
End While
$Y_n^c = Y_{j+1}^c$ % *Corrected end-of-slice value of the solution*
$s_n^c = s_{j+1}^c$ % *End-of-slice value of the rescaled time*
$\Delta T_n^c = \beta_n^p s_n^c$ % *Real size of the time-slice*

Chapter 10

Appendix 2: Some Features of Parallel Implementation

Communication Cost

The cost of sending a message can be represented by two parameters: the message startup time t_s, which is the time required to initiate a communication, and the transfer time t_n required to send a message made of only one number. The time required to send a message of size L numbers is then:

$$T_{msg} = t_s + Lt_n.$$

Therefore, the communication cost depends on the number of messages and on their sizes and is described below for the communication step occurring after each k^{th} iteration.

1. Each processor p sends to a master processor:

   - the number $n_{div}^{(k)}(p)$:
     The sent data is a single number, of size $L_1 = 1$.
   - The sizes $\{\Delta T_n^c = \beta_n^p s_n^c\}(p)$ of its slices (i.e. slices number n, where $n \in \{n_s^{(k-1)} + 1, \cdots, n_s^{(k)}\}$ and $n - n_s$ congruent to p modulo n_p).
     The sent data is of size equal to the number of slices that have been solved by processor p, that is averaged by:
     $L_1' = \frac{1}{n_p}\left[\max_p n_{div}^{(k)}(p) - n_s^{(k-1)}\right] \approx \frac{1}{n_p}\left[n_{conv^{(k)}} + n_p\right]$

   The total number of sent messages is $n_1 = n_p - 1$.

2. The master processor sends to all other processors:

   - the number $n_s^{(k)}$ of the last slice having globally converged to the other processors.
     The sent data is a single number, of size $L_2 = 1$.
   - the starting times $\{T_n^c\}$, for $n_s^{(k-1)} + 1 \leq n \leq n_s^{(k)}$.
     The sent data is of size $L_2' = n_{conv}^{(k)}$.

   The total number of sent messages is $n_2 = n_p - 1$.

3. Every processor having one of the last n_{prec} slices sends to all processors a vector containing the end-of-slice value of the solution, i.e. a *vector having the dimension*

K of the problem, of size $L_3 = K$.
The total number of sent messages is $n_3 = (n_p - 1)n_{prec}$.

The communication cost of the communication step that occurs after each k^{th} iteration is then the sum of the 3 following terms:

$$\begin{aligned}
T^{(k)}_{comm_1} &= n_1 \left[2t_s + (L_1 + L'_1)\, t_n\right] \\
&= (n_p - 1)\left[2t_s + \left(1 + \tfrac{1}{n_p}\left[n^{(k)}_{conv} + n_p\right]\right) t_n\right] \\
&= (n_p - 1)\left[2t_s + \left(2 + \tfrac{n^{(k)}_{conv}}{n_p}\right) t_n\right] \\
T^{(k)}_{comm_2} &= n_2 \left[2t_s + (L_2 + L'_2)\, t_n\right] \\
&= (n_p - 1)\left[2t_s + \left(1 + n^{(k)}_{conv}\right) t_n\right] \\
T^{(k)}_{comm_3} &= n_3 \left[ts + L_3 t_n\right] \\
&= n_{prec}(n_p - 1)\left[ts + Kt_n\right].
\end{aligned}$$

This yields, over the n_I iterations, a communication time that is the sum of the following 3 sums:

$$\begin{aligned}
T_{comm_1} &= \textstyle\sum_{k=1}^{n_I} T^{(k)}_{comm_1} = \sum_{k=1}^{n_I} (n_p - 1)\left[2t_s + \left(2 + \frac{n^{(k)}_{conv}}{n_p}\right) t_n\right] \\
&= 2n_I\,(n_p - 1)\, t_s + 2n_I\,(n_p - 1)\, t_n + \tfrac{n_p - 1}{n_p} t_n \textstyle\sum_{k=1}^{n_I} n^{(k)}_{conv} \\
&= 2n_I\,(n_p - 1)\, t_s + 2n_I\,(n_p - 1)\, t_n + \tfrac{n_p - 1}{n_p} t_n\,(N - n_s)\,. \\
T_{comm_2} &= \textstyle\sum_{k=1}^{n_I} T^{(k)}_{comm_2} = \sum_{k=1}^{n_I} (n_p - 1)\left[2t_s + \left(1 + n^{(k)}_{conv}\right) t_n\right] \\
&= 2n_I\,(n_p - 1)\, t_s + n_I\,(n_p - 1)\, t_n + (n_p - 1)\, t_n \textstyle\sum_{k=1}^{n_I} n^{(k)}_{conv} \\
&= 2n_I\,(n_p - 1)\, t_s + n_I\,(n_p - 1)\, t_n + (n_p - 1)\, t_n\,(N - n_s)\,. \\
T_{comm_3} &= \textstyle\sum_{k=1}^{n_I - 1} T^{(k)}_{comm_3} = \sum_{k=1}^{n_I - 1} n_{prec}(n_p - 1)\left[ts + Kt_n\right] \\
&= (n_I - 1)\, n_{prec}(n_p - 1)\left[t_s + Kt_n\right].
\end{aligned}$$

Thus, the total communication time, after n_I iterations, is
$T_{comm} = T_{comm_1} + T_{comm_2} + T_{comm_3}$, i.e.:

$$\begin{aligned}
T_{comm} &= 4n_I\,(n_p - 1)\, t_s + 3n_I\,(n_p - 1)\, t_n + (n_p - 1)\left(\tfrac{1}{n_p} + 1\right) t_n\,(N - n_s) \\
&\qquad + (n_I - 1)\, n_{prec}(n_p - 1)\left[t_s + Kt_n\right].
\end{aligned}$$

$$\begin{aligned}
T_{comm} &= n_I\,(n_p - 1)\left[4t_s + 3t_n + n_{prec}\,(t_s + Kt_n)\right] \\
&\qquad + (n_p - 1)\left[\left(\tfrac{1}{n_p} + 1\right) t_n\,(N - n_s) - n_{prec}\,(t_s + Kt_n)\right].
\end{aligned} \tag{10.1}$$

For a problem of a given size K, that needs to be solved on N time-slices and for which the predictive model uses n_{prec} values, expression (10.1) shows clearly how T_{comm} increases with the number n_p of processors and with the number n_I of iterations.

Speed-up and Efficiency Evaluation

Let T_s be the sequential execution time of the algorithm on a single processor and T_{n_p} its parallel execution time on n_p identical processors.
In all the tables of this thesis, T_s is the wall clock time of the sequential implementation where there is no predictions, ratio computations, corrections, nor iterations. This sequential implementation is simply the application of the rescaling method, using the same numerical integration method, on the total number of slices considered in the parallel execution.
T_{n_p} is the wall clock time of the parallel implementation.

Let N be the total number of slices globally solved: $N = N^{(n_I)} = n_s^{(n_I)}$.
Knowing that $\alpha = \frac{n_s}{N}$ is the sequential fraction of RaPTI algorithm that cannot be done in parallel, and assuming that the time needed for solving one slice is $\frac{T_s}{N}$, for all processors, then T_{n_p} can be expressed, in terms of the sequential execution time T_s, as:

$$T_{n_p} = \alpha T_s + (1-\alpha)\frac{T_s}{n_p} + T_{overhead},$$

where $T_{overhead}$ is the total overhead due to the parallelization, i.e. the total time spent by all processors doing work that is not done by the sequential algorithm.
This is mainly due to:

1. The computation of the ratios and the predictions, yielding an overhead T_{pred} depending on the chosen model, on the number of remaining slices at each iteration and on the total number n_I of iterations.

2. The repetition of the integration of one non-converging slice at the end of every iteration, by almost all processors and under the assumption that the divergence occurs on consecutive slices. This yields an overhead equal to $\frac{T_s}{N}$, at each iteration (except for the last one), making:

$$T_{corr} = (n_I - 1)\frac{T_s}{N}. \tag{10.2}$$

3. The communication time T_{comm}, given in (10.1).

The total overhead is then:

$$T_{overhead} = T_{pred} + T_{corr} + T_{comm}. \tag{10.3}$$

For a convenience purpose, we let:

$$\epsilon_{overhead} = \frac{T_{overhead}}{T_s} = \frac{T_{pred} + T_{corr} + T_{comm}}{T_s}.$$

This makes T_{n_p} equal to:

$$T_{n_p} = \frac{n_s}{N}T_s + (1 - \frac{n_s}{N})\frac{T_s}{n_p} + \epsilon_{overhead}T_s. \tag{10.4}$$

The parallel system's speed-up S_{n_p} is the ratio of the sequential execution time to the parallel execution time on n_p processors: $S_{n_p} = \frac{T_s}{T_{n_p}}$, or more explicitly:

$$S_{n_p} = \frac{1}{\frac{n_s}{N} + (1 - \frac{n_s}{N})\frac{1}{n_p} + \epsilon_{overhead}}. \tag{10.5}$$

The maximum speed-up is evaluated by letting the total overhead be equal to zero:

$$S_{n_p}^{max} = \frac{1}{\frac{n_s}{N} + \frac{1}{n_p}\left(1 - \frac{n_s}{N}\right)}. \tag{10.6}$$

meeting exactly *Amdahl's law* [68]. However, and even in the best case where the convergence occurs after one iteration only, the maximum speed-up could never be reached: there is a minimum overhead due to the first predictions and to the communication of the time-vectors.
Even though, it should be noted that the application of RaPTI algorithm yields, in case of asymptotic similarity, speed-ups that are very close to the maximum speed-up (see chapters 5 and 6). The results are not as good, in case of weak similarity (see chapter 7).
On the other hand, a lower bound of the speed-up, which is actually a speed-down, is obtained by assuming that only one slice converges at each iteration, which is guaranteed since the first slice is solved with an exact starting value. The result would then be to have, for $n > n_s$, one divergent (therefore unnecessarily solved) slice for every solved slice! Then, one has $n_s^{(k)} = n_s^{(k-1)} + 1$ $(\forall k)$, $n_I = N - n_s$, $T_{corr} = \frac{N-n_s}{N} T_s$ and $T_{n_p} = \left(1 + \frac{N-n_s}{N}\right) T_s + T_{pred} + T_{comm}$, where T_{pred} and T_{comm} are quite large, due to the large number of iterations. The speed-down becomes:

$$S_{n_p}^{min} = \frac{T_s}{T_{n_p}} = \frac{1}{\left(1 + \frac{N-n_s}{N}\right) + \frac{T_{pred}+T_{comm}}{T_s}}, \tag{10.7}$$

Under our assumption that the divergence occurs on consecutive slices, this is obviously the worst case! In practice, ratio properties make several slices to converge, at each iteration, yielding good speed-ups.

The parallel system's efficiency E_{n_p} is the ratio of the speed-up to the number of processors:

$$E_{n_p} = \frac{T_s}{n_p \times T_{n_p}} = \frac{N}{n_p n_s + (1 - n_s) + n_p N \epsilon_{overhead}}. \tag{10.8}$$

Parallel Architecture, Programming Model & Language

All the experiments of RaPTI algorithm have been done on a dual quad core Intel Xeon E5410 at 2.33 GHz processors with 8 GB of memory and 2xHD 500 GB each. The hard disks are mirror raid configured. The operating system is windows 2008 server.

We have implemented RaPTI algorithm, in this thesis, using the high-level technical language `MATLAB` (which stands for matrix laboratory) in its R2009a version, together with the Parallel Computing Toolbox of MATLAB, version 4.1.
The parallel programming model is based on a distributed memory paradigm. The processors communicate using the message passing functions of the Parallel Computing Toolbox (*Interlab Communication Within a Parallel Job*), mainly:

numlabs	: Total number of labs operating in parallel on current job
labindex	: Index of this lab
labReceive	: Receive data from another lab
labSend	: Send data to another lab
labSendReceive	: Simultaneously send data to and receive data from another lab
labBroadcast	: Send data to all labs or receive data sent to all labs
labProbe	: Test to see if messages are ready to be received from other lab
labBarrier	: Block execution until all labs reach this call

Notations

The following table lists the main notations of chapter 4, that are used throughout chapters 5 to 7.

$[0, T]$	Given interval of integration
N	Total number of slices to be solved
n_s	Number of slices sequentially solved, before starting the parallel process
n_p	Number of active processors
n_I	Number of iterations
ϵ_{tol}^{τ}	Tolerance used for getting the initial rescaled time-step τ
ϵ_{tol}^{eos}	Tolerance up to which is reached the EOS condition
ϵ_{tol}^{g}	Tolerance on relative differences of gaps
$\epsilon_{tol}^{n_s}$	Tolerance on relative differences of ratios, for getting n_s
T_s	Time (in seconds) needed for solving (S) sequentially
T_{n_p}	Time (in seconds) needed for solving (S) in parallel using n_p processors
E_{n_p}	Efficiency while using n_p processors
S_{n_p}	Speed-up while using n_p processors
$S_{n_p}^{max}$	Maximum speed-up, stated by Amdahl's law, while using n_p processors
S	Preset threshold value
Y_n	Value of the solution at the end of slice n, that is at $t = T_n$.
R_n	Value of the ratio-vector at slice n
^{p}	Superscript designating predicted values
^{c}	Superscript designating corrected values
$^{(k)}$	Superscript designating values associated with the k^{th} iteration
$n_s^{(k)}$	Total number of converged slices at the end of the k^{th} iteration
$n_{conv}^{(k)}$	Number of slices having converged during the k^{th} iteration

Table 10.1: List of Notations

Note that the superscript (k) is used for the predicted ratios $R_n^{p^{(k)}}$ and predicted starting values $Y_n^{p^{(k)}}$ of the solution, in order to specify that those predictions are done at the k^{th} iteration. However, there is no need to use the (k) superscript for the corrected end-of-slice values Y_n^c and for the gaps GAP_n, because they are evaluated only once, since the correction procedure stops as soon as a divergence occurs.

Bibliography

[1] **K.Burrage**. *Parallel methods for initial value problems.* App. Num. Math., 11 (1991) 5-25.

[2] **K.Burrage**. *Parallel and sequential methods for ordinary differential equations.* Oxford University Press (1995).

[3] **J.Nievergelt**. *Parallel methods for integration ordinary differential equations.* Comm. ACM, 7 (1964) 731-733.

[4] **A.Bellen, M.Zennaro.** *Parallel algorithms for initial value problems.* Journal of Computational and applied mathematics, 25 (1989) 341-350.

[5] **A.Bellen, R.Vermeglio, M.Zennaro.** *Parallel ODE-solvers with step-size control.* Journal of Computational and applied mathematics, 31 (1990) 277-293.

[6] **P.Chartier, B.Philippe**. *A parallel shooting technique for solving dissipative ODE's.* Computing, vol.51, n3-4 (1993) 209-236.

[7] **J.Erhel, S.Rault**. *Algorithme parallèle pour le calcul d'orbites.* Techniques et Sciences Informatiques, 19, 5 (2000) 649-673.

[8] **J.Saltz, V.Naik.** *Towards developing robust algorithms for solving PDEs on mimd machines.* Parallel computing, 6, 1 (1988) 19-44.

[9] **D.Womble.** *A time-stepping algorithm for parallel computers.* SIAM J. Sci. Stat. Comput., 11, 5 (1990) 824-837.

[10] **W.Hackbusch.** *Parabolic multigrid method.* In R.Glowinsky and J.L.Lions, editors, Computing Methods in Applied Sciences and Engineering, VI, North-Holland (1984) 189-197.

[11] **S.Vandewalle, E.Van de Velde.** *Space-time Concurrent Multigrid Waveform Relaxation.* Annals of Numerical Mathematics, 1,1-4 (1994) 347-363.

[12] **G.Horton, S.Vandewalle, P.Worley.** *An algorithm with polylog parallel complexity for solving parabolic partial differential equations.* SIAM J. Sci. Comput., 16, 3 (1995) 531-541.

[13] **J.L.Lions, Y.Maday, G.Turinici**. *Résolution d'EDP par un schéma en temps "pararéel".* C.R.Acad.Sci.Paris, t.332, Série 1, (2001) 661-668.

[14] **Y. Maday, G. Bal**. *A parareal time discretization for non-linear pde's with application to the pricing of an American put.* Recent developments in Domain Decomposition Methods (Zurich 2002), Lecture Notes in computational Science and Engineering, 23, Springer (2002) 189-202.

[15] **Ch.Farhat, M.Chandesris**. *Time-decomposed parallel time-integrators.* Int. J. Numer. Meth. Engng, 58 (2003) 1397-1434.

[16] **Y. Maday, G.Turinici**. *A parareal in time procedure for the control of partial differential equation.* C.R. Acad. Sci. Paris., Ser.I 335 (2002) 387-392.

[17] **Y. Maday, G.Turinici**. *A parallel in time approach for quantum control: the parareal algorithm.* International Journal of Quantum Chemistry, Vol 93 (2003) 223228.

[18] **G.Staff**. *Convergence and stability of the parareal algorithm: A numerical and Theoritical investigation.* Preprints Numerics NO.2, Norwegian University of Science and technology, Trundheim, Norway, (2003).

[19] **G.Bal**. *Parallelization in time of stochastic ODE.* (2003) submitted.

[20] **G. Staff & E. Ronquist**. *Stability of the Parareal Algorithm.* In the 15th International Conference on Domain Decomposition Methods, Berlin (2003). Springer, Lecture Notes in computational Science and Engineering.

[21] **G.Bal**. *On the convergence and the stability of the parareal Algorithm to solve Partial Differential Equations.* In the 15th International Conference on Domain Decomposition Methods, Berlin (2003). Springer, Lecture Notes in computational Science and Engineering.

[22] **P.Fisher, F.Hecht, Y.Maday**. *A parareal in time semi-implicit approximation of the Navier-Stokes equations.* In the 15th International Conference on Domain Decomposition Methods, Berlin (2003). Springer, Lecture Notes in computational Science and Engineering, 40: 433-440.

[23] **Y. Maday & G. Turinici**. *The Parareal in Time Iterative Solver: a further direction to parallel implementation.* In the 15th International Conference on Domain Decomposition Methods, Berlin (2003). Springer, Lecture Notes in computational Science and Engineering.

[24] **Garrido-Espedal-Fladmark**. *A convergent algorithm for time parallelization applied to reservoir simulation.* In the 15th International Conference on Domain Decomposition Methods, Berlin (2003). Springer, Lecture Notes in computational Science and Engineering.

[25] **GA Staff**. *The parareal algorithm, a survey of present work.* NOTUR Emerging technologies, Cluster technologies (2003), Norwegian University of Science and Technology, Department of Mathematical Sciences.

[26] **GA Staff**. *Solving the unsteady Navier-Stokes Equation. An introduction to how to handle time.* (2004).

[27] **D.Tromeur-Dervout**. *Adaptative time domain decomposition for systems of ODEs on grid architecture.* In proc. Int. Conf. PCFD04, (2004).

[28] **N.Chandra-A.Srinivasan**. *Latency and fault tolerance through parallelization in time, with applications to nano-materials.* (Florida States University, Department of Mechanical Engineering) (2004).

[29] **J.Gander, S.Vandewalle**. *Analysis of a two-level time-parallel time-integration method.* Technical report, Katholieke Universiteit Leuven, Dept.of Computer Science (2005).

[30] **S. Daoud**. *Stability of the parareal time discretization for parabolic inverse problem.* In the 16th International Conference on Domain Decomposition Methods, NYU (2005). Proceedings.

[31] **D.Guibert, D.Tromeur-Dervout**. *Adaptative Parareal for systems of ODEs.* In the 16th International Conference on Domain Decomposition Methods, NYU (2005). Proceedings.

[32] **C.Lai, D.Crane, A.Davies**. *On a Parallel Time-domain Method for the non-linear Black-Scholes equation.* In the 16th International Conference on Domain Decomposition Methods, NYU (2005). Proceedings.

[33] **S.Vandewalle**. *PARAREAL in a historical perspective (a review of space-time parallel algorithms).* In the 16th International Conference on Domain Decomposition Methods, NYU 2005. Proceedings.

[34] **M.Gander, S.Vandewalle**. *On the superlinear and linear convergence of the parareal algorithm.* In the 16th International Conference on Domain Decomposition Methods, NYU (2005). Proceedings.

[35] **D.Guibert, D.Tromeur-Dervout**. *Adaptive Parallel Extrapolation method for stiff ODEs.* In Proc. Int. Conf. PCFD05 (2005).

[36] **D.Guibert, D.Tromeur-Dervout**. *Parallel adaptative time domain decomposition for stiff systems of ODEs/DAEs.* Computers and Structures, Volume 85, Issue 9 (2007) 553-562.

[37] **J.Cortial, C.Farhat**. *A time-parallel implicit methodology for the near-real-time solution of systems of linear oscillators.* In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Wanders, editors, Real-Time PDE-Constrained Optimization. Springer (2006).

[38] **C.Farhat, J.Cortial, C.Dastillung, H.Bavestello**. *Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses.* Int. J. Numer. Methods Eng., 67(5) (2006) 697724.

[39] **G.Bal, Q.WU**. *Symplectic Parareal.* Domain Decomposition Methods in Science and Engineering XVII, Lecture Notes in Computational Science and Engineering, Volume 60, I, (2008) 401408.

[40] **M.J. Gander, E. Hairer**. *Nonlinear Convergence Analysis for the Parareal Algorithm.* Domain Decomposition Methods in Science and Engineering XVII. Lecture Notes in Computational Science and Engineering, Volume 60, I (2008) 45-56.

[41] **C.E. Schaerer, T. Mathew, M. Sarkis**. *Block Diagonal Parareal Preconditioner for Parabolic Optimal Control Problems.* Domain Decomposition Methods in Science and Engineering XVII. Lecture Notes in Computational Science and Engineering, Volume 60, II, (2008) 409-416.

[42] **J.Cortial, C.Farhat**. *A Time-Parallel Implicit Method for Accelerating the Solution of Nonlinear Structural Dynamics Problems.* Int. J. Numer. Meth. Engng (2008) 0:125

[43] **P.Amodio, L.Brugnano**. *Parallel solution in time of ODEs: some achievements and perspectives.* Applied Numerical Mathematics, Volume 59, N.3-4 (2009) 424-435.

[44] **C.Audouze, M.Massot, S.Volz**. *Symplectic multi-step parareal algorithms applied to molecular dynamics.* Laboratoire EM2C, Ecole Centrale Paris, hal-00358459 (2009).

[45] **A.Chorin**. *Estimates of intermittency, spectra and blow-up in developed turbulence.* Commun. Pure Appl. Math. 34 (1981) 853-866.

[46] **B.Le Mesurier, G.Papanicolaou, C.Sulem, P.L.Sulem**. *The focusing singularity of the nonlinear Shrodinger equation.* In Directions in Partial Differential Equations, Editors Crandall M. et Al., Academic Press (1987) 159-201.

[47] **M.Berger, R.Kohn**. *Rescaling algorithm for the numerical calculation of blowing-up solutions.* Commun. Pure Appl. Math. 41 (1988) 841-963.

[48] **N.R.Nassif, D.Fayad, M.Cortas**. *Sliced-Time Computations with Re-scaling for Blowing-Up Solutions to Initial Value Differential Equations.* V.S. Sunderam. et al. (Eds): ICCS 2005, LNCS 3514, Springer-Verlag 2005, 58-65.

[49] **M.Cortas**. *Méthode de re-dimensionnement (re-scaling technique) pour des équations aux dérivées ordinaires du 1er ordre à caractère explosif.* Thèse, Université Reims 1. Janvier 2005.

[50] **N.Makhoul-Karam**. *Résolution numérique d'équations paraboliques semi-linéaires à caractère d'explosion ou d'extinction par des méthodes de re-dimensionnement.* Mémoire de DEA en MSI, 2002-2003.

[51] **N.Nassif, N.Makhoul-Karam, Y.Soukiassian**.*Computation of blowing-up solutions for second-order differential equations using re-scaling techniques.* Journal of Computational and Applied Mathematics, Volume 227, Issue 1 (2009) 185-195.

[52] **P.Chartier**. *Introduction to the theory of Ordinary Differential Equations, Chapter 3, (Solution des équations différentielles linéaires)*, on line lecture notes http://www.irisa.fr/ipso/fichiers/cours3.pdf, (March 2010)

[53] **N.Nassif, N.Makhoul-Karam, Y.Soukiassian**. *A New Approach for Solving Evolution Problems in Time-Parallel Way.* International Conference on Computational Science ICCS 06, Proceedings, Part I. Springer-Verlag Berlin / Heidelberg. Volume 3991 (2006) 148 - 155.

[54] **Y.Soukiassian**. *Parallel Algorithms for Time Integration.* AUB MS Thesis. September 2007.

[55] **Haraux A. *Systèmes dynamiques dissipatifs et applications.* RMA, vol 17, Masson, Paris, Milan, Barcelone, Bonn (1991).**

[56] Lasalle J. *Some extensions of Liapunov's second method, IRE Trans.* Circuit Theory, CT-7 (1960) 520-527.

[57] P.Souplet ***Etude des solutions globales de certaines équations différentielles ordinaires du second ordre non-linéaires.*** **C.R. Acad. Sci. Paris, t., 313, Série I (1991), 365-370.**

[58] P.Souplet. ***Existence of exceptional growing-up solutions for a class of nonlinear second order ordinary differential equations.*** **Asymptotic Analysis, 11 (1995), 185-207.**

[59] P.Souplet ***Critical Exponents, special large-time behavior and oscillatory blow-up in nonlinear ode's*** **Differential and Integral Equations, 11, (1998), 147-167.**

[60] P.Souplet, M.Jazar, M.Balabane ***Oscillatory blow-up in nonlinear second order ode's: The critical case.*** **Discrete And Continuous dynamical systems Volume 9, Number 3, (2003) 577-584.**

[61] P.E.Maingé. ***Resolution numérique d'équations de réaction-diffusion intervenant en physique des plasmas.*** **Thèse, Université Bordeaux 1. Janvier 1996.**

[62] S.Rault ***Algorithmes parallèles pour le calcul d'orbites.*** **Université de Rennes 1. Thèse, Septembre 1998.**

[63] Olivier Zarrouati. ***Trajectoires spatiales.*** **CNES (1987).**

[64] B.Hofmann, W.H.Moritz. ***Physical Geodesy, chapter 7.*** **SpringerWien-NewYork, (2005)**

[65] Luc Duriez. ***Cours de mécanique céleste classique.*** **Laboratoire d'astronomie de l'université de Lilles 1 et IMCCE de l'observatoire de Paris. Révision du 8 septembre 2002.**

[66] E.A. Coddington, N. Levinson. ***Theory of Ordinary Differential Equations.*** **McGraw-Hill, New York (1955).**

[67] V.I.Arnold. ***Ordinary Differential Equations.*** **Springer, (1992).**

[68] G.Amdahl. ***Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities.*** **AFIPS Conference Proceedings (30), (1967) 483485.**

[69] J.L.Gustafson. ***Reevaluating Amdahl's Law.*** **Comm.ACM, Vol.31, 5 (1988) 532-533.**

Il me demanda :

Rêves et réalité peuvent-ils se rejoindre ?

Je lui répondis :

Nous rêvons notre avenir,

et plaçons nos rêves à des instants inconnus...

Puis le temps passe et la réalité progresse,

mêlée aux aléas de notre vie...

Nos rêves sont alors corrigés,

et leurs instants précisés...

Ce processus est réitéré,

jusqu'à la fin de notre parcours...

NMK

VU:

Le Directeur de Thèse

VU:

Le Responsable de l'Ecole Doctorale

VU pour autorisation de soutenance
Rennes, le

Le Président de l'Université de Rennes 1

Guy CATHELINEAU

VU après soutenance pour autorisation de publication:

Le Président de Jury,